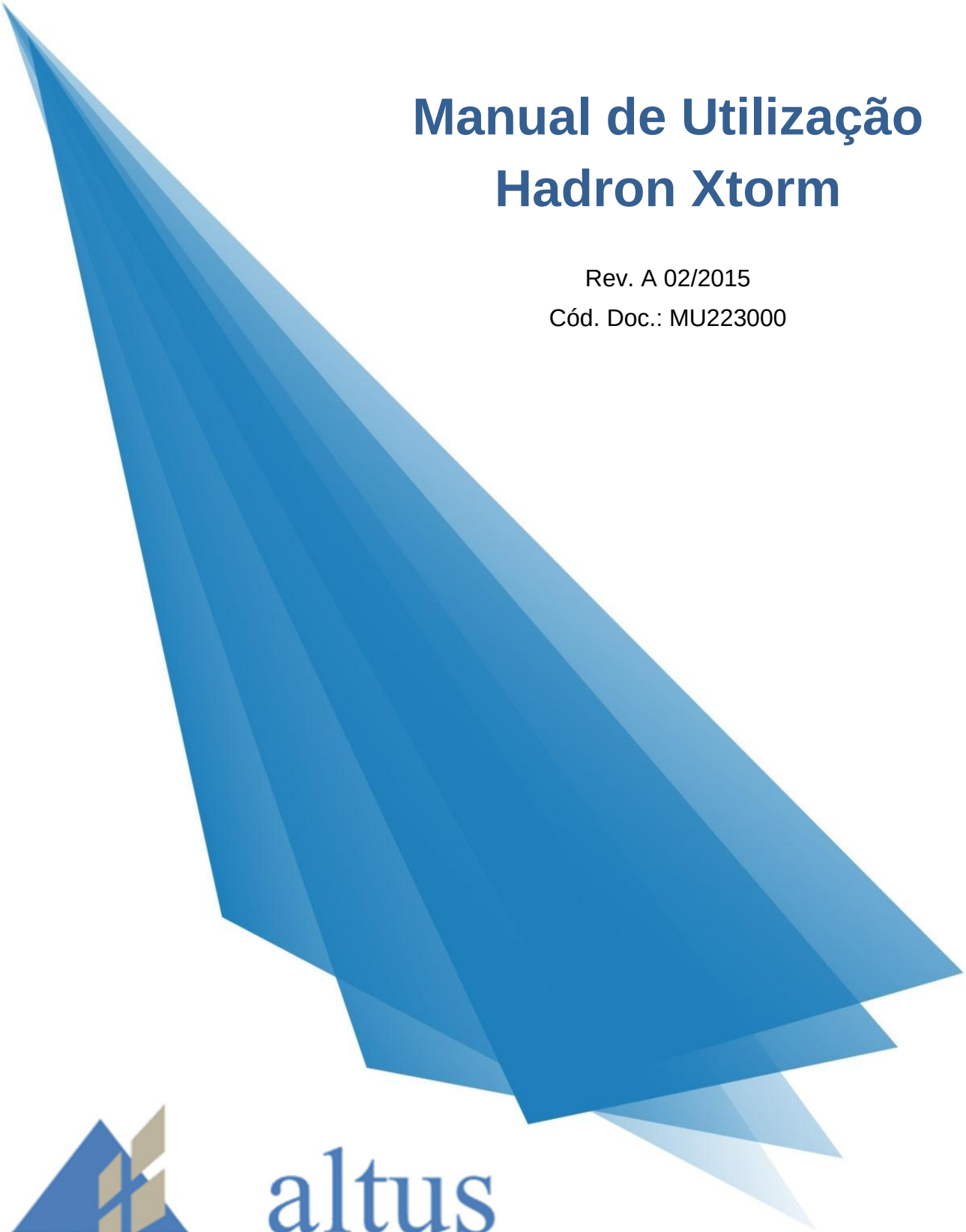
Manual de Utilização
Hadron Xtorm

Rev. A 02/2015

Cód. Doc.: MU223000



altus

www.altus.com.br

Nenhuma parte deste documento pode ser copiada ou reproduzida sem o consentimento prévio e por escrito da Altus Sistemas de Automação S.A., que se reserva o direito de efetuar alterações sem prévio comunicado.

Conforme o Código de Defesa do Consumidor vigente no Brasil, informamos, a seguir, aos clientes que utilizam nossos produtos aspectos relacionados com a segurança de pessoas e instalações.

Os equipamentos de automação industrial fabricados pela Altus são robustos e confiáveis devido ao rígido controle de qualidade a que são submetidos. No entanto, equipamentos eletrônicos de controle industrial (controladores programáveis, comandos numéricos, etc.) podem causar danos às máquinas ou processos por eles controlados em caso de defeito em suas partes e peças ou de erros de programação ou instalação, podendo inclusive colocar em risco vidas humanas.

O usuário deve analisar as possíveis consequências destes defeitos e providenciar instalações adicionais externas de segurança que, em caso de necessidade, sirvam para preservar a segurança do sistema, principalmente nos casos da instalação inicial e de testes.

Os equipamentos fabricados pela Altus não trazem riscos ambientais diretos, não emitindo nenhum tipo de poluente durante sua utilização. No entanto, no que se refere ao descarte dos equipamentos, é importante salientar que quaisquer componentes eletrônicos incorporados em produtos contêm materiais nocivos à natureza quando descartados de forma inadequada. Recomenda-se, portanto, que quando da inutilização deste tipo de produto, o mesmo seja encaminhado para usinas de reciclagem que deem o devido tratamento para os resíduos.

É imprescindível a leitura completa dos manuais e/ou características técnicas do produto antes da instalação ou utilização do mesmo.

Os exemplos e figuras deste documento são apresentados apenas para fins ilustrativos. Devido às possíveis atualizações e melhorias que os produtos possam incorrer, a Altus não assume a responsabilidade pelo uso destes exemplos e figuras em aplicações reais. Os mesmos devem ser utilizados apenas para auxiliar na familiarização e treinamento do usuário com os produtos e suas características.

A Altus garante os seus equipamentos conforme descrito nas Condições Gerais de Fornecimento, anexada às propostas comerciais.

A Altus garante que seus equipamentos funcionam de acordo com as descrições contidas explicitamente em seus manuais e/ou características técnicas, não garantindo a satisfação de algum tipo particular de aplicação dos equipamentos.

A Altus desconsiderará qualquer outra garantia, direta ou implícita, principalmente quando se tratar de fornecimento de terceiros.

Os pedidos de informações adicionais sobre o fornecimento e/ou características dos equipamentos e serviços Altus devem ser feitos por escrito. A Altus não se responsabiliza por informações fornecidas sobre seus equipamentos sem registro formal.

DIREITOS AUTORAIS

Série Ponto, MasterTool, Quark, ALNET e WebPLC são marcas registradas da Altus Sistemas de Automação S.A.

Windows, Windows NT e *Windows Vista* são marcas registradas da Microsoft Corporation.

Este(s) produto(s) utiliza(e) tecnologia EtherCAT® (www.ethercat.org).

NOTIFICAÇÃO DE USO DE SOFTWARE ABERTO

Para obter o código fonte de componentes de software contidos neste produto que estejam sob licença GPL, LGPL, MPL, entre outras, favor entrar em contato através do email opensource@altus.com.br. Adicionalmente ao código fonte, todos os termos da licença, condições de garantia e informações sobre direitos autorais podem ser disponibilizdas sob requisição.

Sumário

1. INTRODUÇÃO 7

Série Hadron Xtorm 7
Características Inovadoras 8
Características Principais 9
Documentos Relacionados a este Manual 10
Suporte Técnico 11
Mensagens de Advertência Utilizadas neste Manual 12

2. DESCRIÇÃO TÉCNICA 13

Painéis e Conexões 13
Características Gerais 15
Características Comuns 15
Características Gerais Específicas 16
Interfaces Seriais 18
Interfaces Ethernet 19
IRIG-B 20
Visor Gráfico 20
Interface do Cartão de Memória 20
Compatibilidade com Demais Produtos 21
Desempenho 21
Tempos de Aplicação 21
Tempos para Execução de Instruções 22
Tempos de Inicialização 22
Produtos Relacionados 22

3. INSTALAÇÃO 25

Inspeção Visual 25
Instalação Mecânica 25
Fixação do Bastidor 25
Inserção dos Módulos 28
Remoção de Módulos 30
Instalação Elétrica 33
Segurança Elétrica 33
Bornes com Mola 34
Conexões 38
Alimentações 38
Fusíveis 38
Instalação Elétrica UCP 38
Conexão com a Rede Ethernet 39
Endereço IP 39
ARP Gratuito 41
Instalação do Cabo de Rede 41
Conexão com a Rede Serial (COM 1) 43
Comunicação RS-232C 43
Comunicação RS-485 sem Terminação 43
Comunicação RS-485 com Terminação Interna 44
Comunicação RS-485 com Terminação Externa 44
Conexão com a Rede Serial (COM 2) 45

Comunicação RS-485 sem Terminação 46
Comunicação RS-485 com Terminação Interna 47
Comunicação RS-485 com Terminação Externa 48
Comunicação RS-422 sem Terminação 48
Comunicação RS-422 com Terminação Interna 49
Exemplo de Rede RS-422 51
Conexão com Sinal IRIG-B 52
Terminais de Ligação do IRIG-B 54
Instalação do Cartão de Memória 55
Instalação do Programador 57

4. CONFIGURAÇÃO 62

Configuração da UCP 62
Parâmetros Gerais 63
Sincronização de Tempo 67
Pontos Internos 70
Conversão de Engenharia 75
Alarmes 76
Agrupamento de Eventos 77
Configuração das Interfaces Seriais 78
COM 1 78
COM 2 81
Configuração das Interfaces Ethernet 84
Interfaces Ethernet Locais 84
Portas TCP Reservadas 84
Configurações Avançadas das Interfaces Ethernet 84
Pontos Duplos 87
Configuração de Protocolos 87
Comportamento dos Protocolos x Estados da UCP 89
Fila de Eventos da UCP 90
Interceptação de Comandos Oriundos do Centro de Controle 93
MODBUS – Tipos de Dados 97
MODBUS RTU Mestre 97
MODBUS RTU Escravo 106
MODBUS Ethernet 112
MODBUS Ethernet Cliente 113
MODBUS Ethernet Servidor 122
Desempenho da Comunicação MODBUS 126
DNP3 – Tipos de Dados 127
DNP3 Ethernet Cliente 128
DNP3 Ethernet Servidor 144
Desempenho da Comunicação DNP3 155
IEC61850 Servidor 156
Desempenho do Sistema 173
Escaneamento de E/S 174
Cartão de Memória 174
Relógio RTC 175
Blocos Funcionais e Funções para Leitura e Escrita do RTC 175
Estruturas de Dados do RTC 179
Memória de Arquivos de Usuário 182
Cartão de Memória 184
Acesso no MasterTool 186
Menu Informativo e de Configuração da UCP 188
Blocos Funcionais e Funções 191

Atualização de Entradas e Saídas 191
Timer Retentivo 193
Timer Não-Redundante 196
Gerenciamento de Usuários e Direitos de Acesso 199
Gerenciamento de Usuários e Direitos de Acesso do Projeto 199
Gerenciamento de Usuários e Direitos de Acesso da UCP 206

5. PROGRAMAÇÃO INICIAL 214

Organização e Acesso à Memória 214
Perfis de Projeto 217
Perfil para UTR 217
Perfil Personalizado 218
Novo Projeto 219
Acrescentando Módulos 223
Criando POUs 225
Criando Tarefas 227
Configurando as Tarefas 228
Vínculo de POUs com Tarefas 230
Número Máximo de Tarefas 230
Configurando a UCP 232
Bibliotecas 233
Inserindo uma Instância de Protocolo 234
MODBUS RTU 234
MODBUS Ethernet 236
Servidor DNP3 238
Localizando a Rede 239
Compilando um Projeto 241
Login 242
Modo Run 244
Modo Stop 245
Monitoração, Escrita e Forçamento de Variáveis 246
Variáveis utilizadas em mais de uma origem 247
Logout 247
Modo Simulação 248
Upload do Projeto 249
Estados de Operação da UCP 251
Run 251
Stop 251
Breakpoint 251
Exception 251
Reset a Quente 251
Reset a Frio 252
Reset Origem 252

6. REDUNDÂNCIA COM UCP HX3040 253

Introdução 253
Configurações para uma UCP Redundante 254
Identificação de uma UCP HX3040 254
Características Gerais 255
Princípios de Funcionamento 257
Projeto Redundante Único 257
Estrutura do Projeto Redundante 257
Mapeamentos Múltiplos 260
Estruturas de Dados de Diagnósticos, Comandos e Usuário 261

Serviços de Sincronização Cíclicos através de Canais de Sincronismo de Redundância 262
Serviços de Sincronização Esporádicos Através dos Canais de Sincronismo de Redundância 263
Redes Ethernet Redundantes com NIC Teaming 264
Métodos de Troca de IP 264
Uso Combinado de NIC Teaming e IP Ativo 265
Estados de uma UCP Redundante 265
Comandos do menu de redundância do visor da UCP 268
Transições entre Estados de Redundância 268
Primeiros Instantes em Estado Ativo 271
Falhas mais Comuns Causadoras de Switchovers Automáticos entre UCPs 272
Falhas Associadas à Switchovers entre UCPs Gerenciados pelo Usuário 272
Tolerância a Falhas 273
Overhead da Redundância 275
Programação de uma UCP Redundante 276
Wizard para Criação de um Novo Projeto Redundante 276
Configuração do projeto com redundância de UCP 281
Configurações das Portas Ethernet da UCP HX3040 (NET 1 até NET 6) 281
Configurações de I/O Drivers 282
Configurações da MainTask 282
Configurações da ProtTask 289
GVLs com Variáveis Simbólicas Redundantes 290
POUs do Tipo Programa com Variáveis Simbólicas Redundantes 290
Utilização de Breakpoints em Sistemas Redundantes 290
Limitações na Programação de uma UCP Redundante 290
Obtendo o Estado da Redundância de uma UCP 291
Leitura de Diagnósticos Não-Redundantes 291
Carga de Programas em uma UCP Redundante 291
Carga Inicial de um Projeto Redundante 292
Conexão do MasterTool com uma UCP HX3040 de uma UCP Redundante 293
Carga de Modificações em um Projeto Redundante 294
Carga de Modificações Off-Line e On-Line 294
Carga On-Line de Modificações 295
Carga Off-Line de Modificações com Interrupção do Controle do Processo 296
Manutenção de UTRs Redundantes 296
Mensagens de Advertência do MasterTool 296
Interação com a Redundância Através do Visor Gráfico da UCP HX3040 297
Estrutura de Diagnósticos da Redundância 297
Logs de Eventos da Redundância 304

7. MANUTENÇÃO 306

Diagnósticos do Módulo UCP 306
One Touch Diag 306
Diagnósticos via LED 309
Diagnósticos via WEB 309
Diagnósticos via Variáveis 314
Diagnósticos via Blocos Funcionais 325
Visor Gráfico 326
Log de Sistema 328
Diagnósticos dos Módulos de E/S 329
Não Carregar a Aplicação na Inicialização 332
Falha na Alimentação 332
Problemas mais Comuns 332
Solução de Problemas 333
Manutenção Preventiva 333

8. PROJETO DO PAINEL ELÉTRICO 334

Projeto Mecânico 334
Dimensões 334
Profundidade do Módulo Montado no Bastidor 334
Espaçamento entre módulos e outros equipamentos do painel 334
Dimensionamento da Calha 335
Montagem Horizontal/ Vertical 335
Projeto Térmico 335
Dissipação de calor em um painel elétrico 336
Projeto Elétrico 339
Informações Gerais 339
Alimentação do Armário 339
Distribuição dos Cabos no Armário 339
Iluminação do Armário 340
Aterramento 340
Interferência Eletromagnética 340
Blindagem 340
Supressores de Ruído 341
Distribuição das Alimentações fora do Armário 343
Proteção contra Raios 343

9. GLOSSÁRIO 345

1. Introdução

Série Hadron Xtorm

A Série Hadron Xtorm consiste na solução no estado-da-arte em Unidades Terminais Remotas (UTRs). A Série apresenta um conjunto ideal de recursos de alto desempenho, facilidades de uso, diagnósticos precisos e completos, design moderno e robusto, além de diversas características inovadoras. A Hadron Xtorm possui uma arquitetura inteligente e versátil, oferecendo modularidade em pontos de entrada e saída (E/S), opções em redundância, protocolos de comunicação de alta velocidade, como IEC 61850, implementação de lógica em conformidade com a norma IEC 61131-3, sendo uma UTR ideal para todas as aplicações de supervisão e controle de sistemas de geração, transmissão e distribuição de energia elétrica, como usinas hidrelétricas (UHEs), e subestações de energia.

A Série Hadron Xtorm possui uma avançada tecnologia em seu barramento que utiliza uma interface Ethernet de alta velocidade possibilitando que informações de entradas, saídas e dados possam ser compartilhados entre múltiplos controladores dentro de um mesmo sistema. O sistema pode ser facilmente dividido e distribuído em todo o campo, possibilitando o uso de expansões de bastidores com o mesmo desempenho de um módulo local permitindo que todos os tipos de módulos sejam utilizados tanto no bastidor local quanto nas expansões de bastidores sem restrições. Para a interligação entre as expansões de bastidores é utilizado um simples cabo padrão Ethernet.

Figura 1-1. Série Hadron Xtorm – Visão Geral

Características Inovadoras

A Série Hadron Xtorm traz aos usuários diversas inovações na utilização, supervisão e manutenção do sistema. Estas características foram desenvolvidas focando um novo conceito em automação de usinas hidrelétricas, subestações e demais aplicações deste segmento. A lista abaixo mostra algumas destas características que o usuário encontrará na Série Hadron Xtorm:

Battery Free Operation: A Série Hadron Xtorm não requer nenhum tipo de bateria para manutenção de memória e operação do relógio de tempo real. Esta funcionalidade é extremamente importante, pois reduz a necessidade de manutenção do sistema e permite o uso em locais remotos de difícil manutenção. Além disto, esta característica é ambientalmente correta.

Multiple Block Storage: Diversos tipos de memória (voláteis e não voláteis) estão disponíveis nas UCPs da Série Hadron Xtorm, oferecendo a melhor opção para cada necessidade. As UCPs da Série Hadron Xtorm oferecem variáveis de entrada de representação direta (%I), variáveis de saída de representação direta (%Q), memória de dados e variáveis simbólicas. Para aplicações que necessitam funcionalidades de memória não volátil, a Série Hadron Xtorm possibilita a utilização de variáveis de representação direta de memória retentiva (%Q), memória retentiva de dados, variáveis simbólicas, memória persistente de dados, memória de programa, memória de código fonte, sistema de arquivos na UCP (arquivos DOC, PDF, imagens, dados, entre outros) e interface para cartão de memória.

One Touch Diag: Esta é uma funcionalidade da Série Hadron Xtorm. Através deste novo conceito, o usuário pode checar as informações de diagnóstico de qualquer módulo do sistema diretamente no visor gráfico da UCP, mediante apenas um toque no botão de diagnóstico do respectivo módulo. A OTD é uma poderosa ferramenta de diagnóstico que pode ser usada offline (sem supervisor ou programador) e reduz os tempos de manutenção e comissionamento.

OFD – On Board Full Documentation: A UCP da Série Hadron Xtorm tem a capacidade de armazenar a documentação completa do projeto na sua memória. Este é um recurso interessante para fins de backup e manutenção, já que a informação completa fica armazenada em um único e seguro local.

ETD – Electronic Tag on Display: Outra característica apresentada pela serie Hadron Xtorm é o ETD. Esta funcionalidade possibilita a verificação do tag de qualquer ponto ou módulo de E/S usado no sistema, diretamente no visor gráfico da UCP. Juntamente com esta informação, o usuário pode também verificar a descrição. Este é um recurso extremamente útil durante a manutenção e resolução de problemas.

Características Principais

UCPs

As UCPs apresentam várias funções integradas, programação online, alta capacidade de memória e canais seriais. Estes dispositivos possuem um moderno processador de 32 bits que proporciona um excelente desempenho e a integração de diversas funcionalidades. A UCP conta com porta Ethernet para programação, uso em redes MODBUS, DNP3 e servidor web embarcado. Apresentam duas interfaces seriais para conexão de IHMs locais e redes MODBUS. As UCPs têm uma posição de cartão de memória para hospedagem das páginas da web, código fonte da aplicação, atualização de firmware entre outras características.

Módulos

Os módulos apresentam E/S de alta densidade. Cada módulo de E/S apresenta um visor para diagnósticos locais no qual é mostrado o estado de cada ponto de E/S. Há também os diagnósticos multifuncionais sobre o status dos módulos. Todas as informações de diagnóstico também podem ser acessadas remotamente pela UCP, protocolos de comunicação ou pela ferramenta de configuração MasterTool Xtorm.

Barramento de Alta Velocidade

A arquitetura da Série Hadron Xtorm apresenta um barramento no estado da arte de barramentos baseados em Ethernet de 100 Mbps. A alta taxa de transferência permite a atualização de grandes quantidades de entradas em um curto período de tempo. Os módulos são endereçados e identificados automaticamente evitando erros durante a configuração da aplicação e manutenção de campo. O barramento fornece características especiais que permitem redundância da UCP no mesmo bastidor entre outras características listadas abaixo.

- Endereçamento e identificação de módulo automática
- Troca a quente
- Barramento serial baseado em Ethernet 100 Mbps
- Sincronização de tempo para atualização de E/S ou estampa de tempo precisa
- Solução de hardware de chip único

Troca a Quente

A característica de troca a quente permite a substituição de módulos sem desenergização do sistema. A UCP mantém o controle de todo o processo e os módulos podem ser substituídos sempre que necessário.

Alta Disponibilidade

A Série Hadron Xtorm oferece diversas arquiteturas diferentes de redundância, onde UCPs, Fontes de Alimentação e Interfaces de Rede podem ser montadas em uma aplicação redundante. Com esta flexibilidade o sistema pode ser ajustado desde sistemas simples sem nenhuma redundância até complexas e críticas aplicações onde a alta disponibilidade é essencial.

Diagnósticos Avançados

Cada módulo contém seus próprios diagnósticos: as UCPs, Interfaces de Rede, Fontes de Alimentação e módulos de E/S apresentam vários diagnósticos disponíveis. Cada módulo tem um visor multifuncional que informa o status do mesmo. Além disto, cada módulo conta com um botão na sua parte frontal para fornecer informações de diagnósticos diferentes para a equipe de manutenção. Estes diagnósticos podem ser monitorados no campo através de visores ou através da ferramenta de configuração. Alguns exemplos são:

- Módulo localizado em uma posição incorreta no bastidor
- Ausência de fonte de alimentação

- Curto circuito nas saídas
- Não há configuração para um módulo que necessite ser configurado para operação normal
- Visualização de tag e descrição de E/S da UTR
- Visualização de endereço IP

Capacidades

Na Série Hadron Xtorm o maior bastidor pode conter até 18 módulos, sendo que devemos tomar o cuidado de que a combinação dentre os módulos escolhidos não ultrapasse o limite de corrente da fonte de alimentação do bastidor. A corrente que cada módulo da Série Xtorm consome do barramento pode ser encontrado no documento com as características técnicas de cada módulo, ou então podemos utilizar a funcionalidade de "Configuração e Consumo" do Mastertool Xtorm para montar a arquitetura com os módulos desejados e visualizar o consumo de corrente de cada módulo, a corrente total necessária para os módulos escolhidos e o valor fornecido pela fonte de alimentação. Com esta arquitetura uma UCP única pode controlar 512 pontos de E/S usando apenas um bastidor. É possível expandir para até 9 bastidores (bastidor principal + 8 bastidores de expansão) utilizando a funcionalidade de expansão de barramento. Neste caso, o limite máximo de módulos contabilizados dentre todos os bastidores de expansão não podem ultrapassar 64 módulos.

Programação da UCP & Atualização de Firmware

A Série Hadron Xtorm permite a programação da UCP e a atualização do firmware através da porta Ethernet da UCP. Esta abordagem oferece algumas funcionalidades, tais como:

- Porta Ethernet multifuncional usada para compartilhar programação, intercâmbio de dados ponto a ponto, protocolo de dispositivo de terceiros na camada de aplicação, intercâmbio de dados das variáveis de rede, etc.
- Acesso direto às variáveis locais da UCP
- Acesso remoto via interface Ethernet
- Atualização de firmware via interface Ethernet

Documentos Relacionados a este Manual

Para obter informações adicionais sobre a Série Hadron Xtorm, podem ser consultados outros documentos (manuais e características técnicas) além deste. Estes documentos encontram-se disponíveis em sua última revisão em http://www.altus.com.br.

Cada produto possui um documento denominado Característica Técnica (CT), onde encontram-se as características do produto em questão.

Aconselham-se os seguintes documentos como fonte de informação adicional:

Código	Descrição	Idioma
CE108804	MasterTool Xtorm Technical Characteristics	Inglês
CT108804	Características Técnicas MasterTool Xtorm	Português
CS108804	Características Técnicas MasterTool Xtorm	Espanhol
CE123000	Hadron Xtorm Series Technical Characteristics	Inglês
CT123000	Características Técnicas Série Hadron Xtorm	Português
CS123000	Características Técnicas Serie Hadron Xtorm	Espanhol
CE123100	CPU 6 ETH, 2 SERIALS, IRIG-B, RED Module Technical Characteristics	Inglês
CT123100	Características Técnicas do Módulo UCP 6 ETH, 2 SERIAIS, IRIG-B, RED.	Português
CS123100	Características Técnicas del UCP 6 ETH, 2 SERIALES, IRIG-B, RED.	Espanhol
CE123701	Hadron Xtorm Series Backplane Racks Technical Characteristics	Inglês
CT123701	Características Técnicas dos Bastidores da Série Hadron Xtorm	Português
CS123701	Características Técnicas de los Bastidores de la Serie Hadron Xtorm	Espanhol

CE123200	Redundant Power Supply 60 W 125 Vdc Module Technical Characteristics	Inglês
CT123200	Características Técnicas do Módulo Fonte de Alimentação Redundante 60 W 125 Vdc	Português
CS123200	Características Técnicas del Módulo Fuente de Alimentación Redundante 60 W 125 Vdc	Espanhol
CE123300	Módulo 32 DI 125 Vdc w/ event log Module Technical Characteristics	Inglês
CT123300	Características Técnicas do Módulo 32 ED 125 Vdc c/ registro de eventos	Português
CS123300	Características Técnicas del Módulo 32 ED 125 Vdc c/ registro de eventos	Espanhol
CE123400	16 SD Relay 125 Vdc w/ CBO Module Technical Characteristics	Inglês
CT123400	Características Técnicas do Módulo 16 SD Relé 125 Vdc c/ CBO	Português
CS123400	Características Técnicas del Módulo 16 SD Relé 125 Vdc c/ CBO	Espanhol
CE123310	16 AI Voltage/Current Module Technical Characteristics	Inglês
CT123310	Características Técnicas do Módulo 16 EA Tensão/Corrente	Português
CS123310	Características Técnicas del Módulo 16 EA Tensión/Corriente	Espanhol
CE123311	8 AI RTD Module Technical Characteristics	Inglês
CT123311	Características Técnicas do Módulo 8 EA RTD	Português
CS123311	Características Técnicas del Módulo 8 EA RTD	Espanhol
CE123901	Hadron Xtorm Series Connectors Technical Characteristics	Inglês
CT123901	Características Técnicas dos Conectores da Série Hadron Xtorm	Português
CS123901	Características Técnicas de los Conectores de la Serie Hadron Xtorm	Espanhol
CE123900	Rack Connector Cover Technical Characteristics	Inglês
CT123900	Características Técnicas da Tampa para conector de bastidor	Português
CS123900	Características Técnicas de la Tapa para conector de bastidor	Espanhol
MU223600	Hadron Xtorm Utilization Manual	Inglês
MU223000	Manual de Utilização Hadron Xtorm	Português
MU223300	Manual de Utilización Hadron Xtorm	Espanhol
MU223601	Hadron Xtorm DNP3 Server Device Profile Document	Inglês
MU223602	Hadron Xtorm DNP3 Client Device Profile Document	Inglês
MU223605	Hadron Xtorm IEC 61850 Device Profile Document	Inglês
MP399609	IEC 61131 Programming Manual	Inglês
MP399048	Manual de Programação IEC 61131	Português
MP399800	Manual de Programación IEC 61131	Espanhol

Tabela 1-1. Documentos Relacionados

Suporte Técnico

Para entrar em contato com o Suporte Técnico da Altus em São Leopoldo, RS, ligue para +55 51 3589-9500. Para conhecer os centros de Suporte Técnico da Altus existentes em outras localidades, consulte nosso site (http://www.altus.com.br) ou envie um e-mail para altus@altus.com.br.

Se o equipamento já estiver instalado, tenha em mãos as seguintes informações ao solicitar assistência:

- Os modelos dos equipamentos utilizados e a configuração do sistema instalado
- O número de série do equipamento
- A revisão do equipamento e a versão do software executivo, constantes na etiqueta afixada na lateral do produto
- Informações sobre o modo de operação do módulo, obtidas através do MasterTool Xtorm
- O conteúdo do programa aplicativo, obtido através do programador MasterTool Xtorm
- A versão do programador utilizado

Mensagens de Advertência Utilizadas neste Manual

Neste manual, as mensagens de advertência apresentarão os seguintes formatos e significados:

PERIGO:
Relatam causas potenciais, que se não observadas, levam a danos à integridade física e saúde, patrimônio, meio ambiente e perda da produção.

CUIDADO:
Relatam detalhes de configuração, aplicação e instalação que devem ser seguidos para evitar condições que possam levar a falha do sistema e suas consequências relacionadas.

ATENÇÃO:
Indicam detalhes importantes de configuração, aplicação ou instalação para obtenção do máximo desempenho operacional do sistema.

2. Descrição Técnica

Este capítulo apresenta todas as características técnicas da UCP da Série Hadron Xtorm HX3040.

Painéis e Conexões

A figura a seguir mostra o painel frontal da UCP HX3040.

Figura 2-1. UCP HX3040

Como se pode observar na figura, na parte superior do painel frontal se encontra o visor gráfico utilizado para mostrar o estado e diagnósticos de todo o sistema, incluindo os diagnósticos específicos de cada módulo. O visor gráfico também oferece um menu fácil de usar que traz ao usuário um modo rápido para ler ou definir alguns parâmetros como: temperatura interna (somente leitura), contraste do visor gráfico, endereço IP para cada interface de rede (somente leitura) e hora local (somente leitura).

A UCP da Série Hadron Xtorm apresenta um visor gráfico contendo o status e informações úteis para o usuário, tais como: estados da aplicação (em execução e parado), status do cartão miniSD, atividade nas interfaces seriais (RX e TX) entre outras. Adicionalmente, as UCPs da Série Hadron Xtorm também fornecem um LED bicolor usado para indicar os status e diagnósticos. Para maiores informações a respeito do significado do status do LED de diagnósticos, consulte o documento com as características técnicas da UCP – CT123100.

No painel frontal estão disponíveis as interfaces de conexão da UCP da Série Hadron Xtorm. Essas interfaces são de comunicação Ethernet, comunicação serial, interface do cartão de memória e interface de sincronismo de tempo. A Tabela 2-1 apresenta uma breve descrição dessas interfaces.

Interfaces	Disponível nos modelos	Descrição
NET 1 ao NET 6	HX3040	Conector tipo RJ45 de comunicação no padrão 10/100Base-TX. Permite a comunicação ponto a ponto ou em rede nos protocolos abertos, MODBUS TCP cliente e servidor, MODBUS RTU via TCP cliente e servidor, Cliente DNP3 e Servidor. Para obter mais informações sobre utilização, consulte o capítulo Configuração – Configuração das Interfaces .
COM 1	HX3040	Conector tipo DB9 fêmea para comunicação no padrão RS-232C e RS-485. Permite a comunicação ponto a ponto ou em rede nos protocolos abertos, MODBUS RTU escravo ou MODBUS RTU mestre.
COM 2	HX3040	Conector tipo DB9 fêmea para comunicação nos padrões RS-485 e RS-422. Permite a comunicação ponto a ponto ou em rede nos protocolos abertos, MODBUS RTU escravo ou MODBUS RTU mestre.
MEMORY SLOT	HX3040	Conector para cartão de memória. Permite utilizar um cartão de memória para diferentes tipos de armazenamento de dados como: logs de usuário, páginas Web, a documentação do projeto e arquivos de origem. Para obter mais informações sobre utilização, consulte o capítulo Configuração – Cartão de Memória.
IRIG-B	HX3040	Borne conector com 4 terminais para comunicação no padrão IRIG-B.

Tabela 2-1. Interfaces de Conexão

Características Gerais

Características Comuns

	HX3040
Linguagens de programação	Lista de Instruções (IL) Texto Estruturado (ST) Diagrama Ladder (LD) Sequenciamento Gráfico de Funções (SFC) Diagrama de Blocos Funcionais (FBD) Gráfico Funcional Contínuo (CFC)
Tipos de tarefas	Cíclica (periódica) Disparada por evento (interrupção de software) Disparado por evento externo (interrupção de hardware) Contínua (execução livre) Disparada por estado (interrupção de software)
Alterações online	Sim
Suporte a troca a quente	Sim
Suporte a redundância de expansão de barramento	Sim
Interfaces seriais	1 x RS-232C / RS-485 (COM 1) 1 x RS-485 / RS-422 (COM 2)
Protocolo MODBUS	Mestre e escravo RTU (COM 1 e COM 2) Cliente e servidor TCP (NET 1 ao NET 6) Cliente e servidor RTU via TCP (NET 1 ao NET 6)
Protocolo DNP3	Cliente e servidor TCP (NET 1 ao NET 6)
Protocolo MMS	Servidor TCP (NET 1 ao NET 6)
Relógio de tempo real (RTC)	Sim Resolução de 1 ms e variação máxima de 2 s por dia
Cão-de-guarda	Sim
Indicação de status e diagnóstico	Visor gráfico, LED, páginas web e memória interna da UCP.
One Touch Diag (OTD)	Sim
Electronic Tag on Display (ETD)	Sim
Isolação	
Lógica para terra de proteção ⏚	2500 Vac / 1 minuto
Lógica para interfaces Ethernet	1500 Vac / 1 minuto
Lógica para porta serial (COM 2)	2000 Vac / 1 minuto
Lógica para porta IRIG-B	2500 Vac / 1 minuto
Interfaces Ethernet para terra de proteção ⏚	1500 Vac / 1 minuto
Interfaces Ethernet para porta serial (COM 2)	2500 Vac / 1 minuto
Interface Ethernet para Interface Ethernet	1500 Vac / 1 minuto
Porta Serial (COM 2) para terra de proteção ⏚	2500 Vac / 1 minuto
Consumo de corrente do bastidor	1500 mA
Dissipação	7,5 W
Temperatura de operação	-5 a 60 °C
Temperatura de armazenamento	-25 a 85 °C
Umidade relativa de operação e armazenamento	5 a 96 %, sem condensação.
Revestimento de circuitos eletrônicos	Sim
Índice de proteção	IP 20
Padrões	IEC 61131-2 IEC 61131-3 IEC 61850 CE, diretivas de Compatibilidade Eletromagnética (EMC) e Dispositivos de Baixa Tensão (Low-Voltage Directive – LVD).

	CE
Diretiva RoHS	RoHS 2002/95/EC
Dimensões do módulo (L x A x P)	38,0 x 235,3 x 184,2 mm
Dimensões da embalagem (L x A x P)	55,0 x 308,0 x 266,0 mm
Peso	1000 g
Peso com embalagem	1300 g

Tabela 2-2. Características Comuns

Notas:

Tipos de tarefas: Tarefa é um objeto usado para chamar POUs. Uma tarefa pode ser disparada por período, eventos ou pode ainda ser executada no modo contínuo. Cada tarefa pode chamar uma ou mais POUs.

Relógio de tempo real (RTC): O tempo de retenção, tempo em que o relógio de tempo real continuará a atualizar a data e hora após a desenergização da UCP, é 15 dias para operação a 25 °C. Na temperatura máxima do produto o tempo de retenção é reduzido para 10 dias.

One Touch Diag (OTD): Essa opção somente estará disponível ao usuário quando o módulo estiver em modo operacional.

Isolação: Lógica é o nome para os circuitos internos como processador, memória e interfaces com o barramento.

Revestimento de circuitos eletrônicos: O revestimento de circuitos eletrônicos protege as partes internas do produto contra umidade, poeira e outros elementos agressivos a circuitos eletrônicos.

Características Gerais Específicas

	HX3040
Memória de variáveis de entrada de representação direta (%I)	96 Kbytes
Memória de variáveis de saída de representação direta (%Q)	96 Kbytes
Memória de variáveis simbólicas	6 Mbytes
Memória de variáveis simbólicas retentivas	8 Kbytes
Memória de variáveis simbólicas persistentes	4 Kbytes
Memória de variáveis redundantes	512 Kbytes
Memória de programa	12 Mbytes
Memória de código fonte (backup)	100 Mbytes
Memória de arquivos de usuário	32 Mbytes
Número máximo de tarefas	32
Número máximo de barramento de expansão	8
Ethernet TCP/IP interface local	6
Suporte a redundância de interfaces Ethernet TCP/IP	Sim
Suporte a redundância de UCP (mesmo bastidor)	Sim
Sincronização do relógio (SNTP)	Sim

Tabela 2-3. Características Gerais Específicas

Notas

Memória de variáveis de entrada de representação direta (%I): Área onde são alocadas as variáveis de representação direta para o tipo entrada. Variável de representação direta significa que a variável pode ser acessada diretamente na memória utilizando o endereço desejado. Por exemplo: %IB0, %IW100. Variável de entrada de representação direta pode ser utilizada para mapear pontos de entrada analógicos ou digitais. Como referência, 8 pontos de entrada digital podem ser representados por um octeto e um ponto de entrada analógica pode ser representado por dois octetos.

A UCP HX3040 da Série Hadron Xtorm define toda área de memória de variáveis de entrada de representação direta %I como variáveis redundantes, ou seja, o usuário não precisa selecionar tal área.

Memória de variáveis de saída de representação direta (%Q): Área onde são alocadas todas as variáveis de representação direta para o tipo saída. Variável de representação direta significa que a variável pode ser acessada diretamente na memória utilizando o endereço desejado. Por exemplo: %QB0, %QW100. Variável de saída de representação direta pode ser utilizada para mapear pontos de saída analógicos ou digitais. Como referência, 8 pontos de saída digital podem ser representados por um octeto e um ponto de saída analógica pode ser representado por dois bytes.

A UCP HX3040 da Série Hadron Xtorm define toda área de memória de variáveis de saída de representação direta %Q como variáveis redundantes, ou seja, o usuário não precisa selecionar tal área.

Memória de variáveis simbólicas: Área onde são alocadas as variáveis simbólicas. As variáveis simbólicas são variáveis IEC criadas em POUs e GVLs durante o desenvolvimento do aplicativo, as quais não são endereçadas diretamente na memória. Variáveis simbólicas podem ser definidas como retentivas ou persistentes, neste caso serão utilizadas as áreas de memória de variáveis simbólicas retentiva ou memória de variáveis simbólicas persistentes respectivamente. O sistema da UTR aloca variáveis de sistema nesta área, desta forma o espaço disponível para a alocação de variáveis criadas pelo usuário é inferior ao informado na tabela. A quantidade de memória ocupada por estas variáveis de sistema depende das características do projeto (quantidade de módulos, de drivers, etc...), desta forma recomenda-se observar o espaço disponível nas mensagens de compilação da ferramenta MasterTool Xtorm.

Memória de variáveis simbólicas retentivas: Área onde são alocadas as variáveis simbólicas retentivas. Os dados retentivos mantêm seus respectivos valores mesmo após um ciclo de desenergização e energização da UCP. A lista completa de quando as variáveis retentivas mantém seus valores e quando o valor é perdido, pode ser encontrada na próxima tabela.

Memória de variáveis simbólicas persistentes: Área onde são alocadas as variáveis simbólicas persistentes. Os dados persistentes mantêm seus respectivos valores mesmo após o download de uma nova aplicação na UCP.

ATENÇÃO: A declaração e utilização de variáveis persistentes deve ser realizada única e exclusivamente através do objeto Persistent Vars, o qual pode ser incluído no projeto através da treeview em Application -> Add Object -> Persistent Variables. Não deve ser utilizada a expressão VAR PERSISTENT no campo de declaração de variáveis das POUs.

Além do tamanho de área persistente informado na tabela acima, estão reservados estes 44 bytes para armazenar informações sobre as variáveis persistentes (não disponível para uso). A lista completa de quando as variáveis persistentes mantém seus valores e quando o valor é perdido, pode ser encontrada na tabela a seguir.

A tabela mostra o comportamento das variáveis simbólicas, retentivas e persistentes para diferentes situações, onde "-" significa que o valor é perdido e "X" significa que o valor é mantido

Condição	Variável simbólica	Variável retentiva	Variável persistente
Reset Quente/Ciclo de	-	X	X

energização			
Reset Frio	-	-	X
Reset Origem	-	-	-
Remover UCP ou Fonte de Alimentação do Bastidor enquanto energizado	-	-	-
Download	-	-	X
Alteração online	X	X	X
Reiniciar UCP	-	X	X
Clean All	-	-	X

Tabela 2-4. Comportamento das Variáveis Não Voláteis

No caso do comando de Clean All, caso a aplicação tenha sido modificada de tal forma que variáveis persistentes tenham sido removidas, inseridas no início da lista ou então tenham tido o seu tipo modificado, o valor destas variáveis será perdido (alertado pela ferramenta MasterTool ao realizar o download). Desta forma recomenda-se que alterações na GVL de variáveis persistentes envolvam somente a inclusão de novas variáveis no final da lista.

Memória de variáveis redundantes: Apenas aplicado para projetos com redundância de UCP. São as variáveis sincronizadas entre as UCPs redundantes, o que contempla as variáveis de representação direta (%I e %Q), utilizadas por módulos de E/S, e variáveis simbólicas do programa do usuário, declaradas em GVLs ou POUs redundantes.

Memória de programa: Área da memória que corresponde ao tamanho máximo permitido para a aplicação de usuário. Essa área é compartilhada com a memória de código fonte, sendo a área total a soma de "memória de programa" e "memória de código fonte".

Memória de código fonte (backup): Área da memória utilizada como backup do projeto, ou seja, caso o usuário deseje importar o seu projeto, o software MasterTool Xtorm irá buscar as informações necessárias nessa área. É importante garantir que o projeto salvo como backup está atualizado para evitar perda de informações críticas. Essa área é compartilhada com a memória de programa sendo a área total a soma de "memória de programa" e "memória de código fonte".

Memória de arquivos de usuário: Essa área da memória é destinada ao armazenamento de arquivos, como: doc, pdf, imagens, entre outros, ou seja, permite a gravação de dados como se fosse um cartão de memória.

Suporte a redundância (mesmo bastidor): A UCP HX3040 suporta redundância de UCP localizadas no mesmo bastidor. Para maiores informações consulte o subcapítulo do Manual de Utilização Hadron Xtorm – MU223000.

Interfaces Seriais

COM 1

	HX3040
Conector	DB9 fêmea blindado
Meio físico	RS-232C ou RS-485 (dependendo do cabo conectado)
Direção da comunicação	RS-232C: full duplex RS-485: half duplex
Número máximo de transceivers RS-485	32
Terminação RS-485	Não (possibilita o uso de terminação ativa externa)
Sinais de modem	RTS, CTS, DCD
Baud rate	600, 1.200, 1.800, 2.400, 4.800, 9.600, 19.200, 38.400, 57.600, 115.200 bps
Protocolos	MODBUS RTU (mestre/escravo)

	Protocolo aberto

Tabela 2-5. Interface Serial COM 1

COM 2

	HX3040
Conector	DB9 fêmea blindado
Meio físico	RS-422 ou RS-485 (dependendo do cabo selecionado)
Direção de comunicação	RS-422: full duplex RS-485: half duplex
Número máximo de transceivers RS-422	11 (1 transmissor e 10 receptores)
Número máximo de transceivers RS-485	32
Terminação	Sim (opcional via seleção de cabo)
Baud rate	600, 1.200, 1.800, 2.400, 4.800, 9.600, 19.200, 38.400, 57.600, 115.200 bps
Protocolos	MODBUS RTU mestre/escravo Protocolo aberto

Tabela 2-6. Interface Serial COM 2

Notas

Meio físico: Dependendo da configuração do cabo usado, é possível escolher o tipo de interface física: RS-232C ou RS-485 para a COM1, e RS-422 ou RS-485 para a COM2.

Número máximo de transceivers RS-422 : Referem-se ao número máximo de interfaces RS-422 que podem ser usadas no mesmo barramento.

Número máximo de transceivers RS-485: Refere-se ao número máximo de interfaces RS-485 que podem ser usadas no mesmo barramento.

Interfaces Ethernet

NET 1 à NET 6

	HX3040
Conector	RJ45 fêmea blindado
Auto negociação (auto crossover)	Sim
Comprimento máximo do cabo	100 m
Tipo de cabo	UTP ou ScTP, categoria 5
Baud rate	10/100 Mbps
Camada física	10/100Base-TX
Camada de enlace de dados	LLC (controle de enlace lógico)
Camada de rede	IP (protocolo de internet)
Camada de transporte	TCP (protocolo de controle de transmissão) UDP (protocolo de datagrama de usuário)

Camada de aplicação	Cliente / Servidor MODBUS TCP Cliente / Servidor MODBUS RTU via TCP Cliente / Servidor DNP3 Protocolo GOOSE (envio e recebimento de mensagens) Servidor MMS HTTP (servidor web) Protocolo de programação Mastertool Xtorm (apenas NET 1) SNTP (sincronismo de relógio)

Tabela 2-7. Interface Ethernet NET 1

Nota:

Protocolo de programação Mastertool Xtorm: Nos casos em que a NET 1 estiver configurada como redundante, a NET 2 também poderá ser utilizada para o Protocolo de programação Mastertool Xtorm.

IRIG-B

	HX3040
Tipo de Conector	Conector borne extraível com 4 terminais (HX9405)
Comprimento máximo do cabo	5 m
Seção do fio	0,5 mm²
Nível de entrada e saída	TTL
Impedância de entrada	> 100 kΩ
Atraso entre entrada e saída	< 10 ns
Corrente máxima na saída	10 mA
Carga máxima na saída	500 Ω
Proteção contra curto circuito	Sim
Níveis de tensão	0 a 1,5 Vdc para nível lógico 0 3,5 a 5 Vdc para nível lógico 1

Tabela 2-8. Interface IRIG-B

Notas:

Carga máxima na saída: A carga total resultante de todos os dispositivos conectados à saída não deve ultrapassar este valor. Não há um limite máximo de dispositivos pré-definido. O mesmo deve ser calculado a partir da impedância de entrada mínima de cada um dos dispositivos conectados à saída IRIG-B da Xtorm.

Visor Gráfico

As UCPs da Série Hadron Xtorm possuem um visor gráfico usado para mostrar o status e diagnóstico do sistema inteiro, inclusive os diagnósticos específicos de cada módulo adicional. O visor também oferece um menu de fácil utilização que dá ao usuário um rápido caminho para ler ou definir alguns parâmetros como: temperatura interna (somente leitura); contraste do visor gráfico e endereço IP para cada interface NET (somente leitura). Mais informações sobre como usar o visor gráfico podem ser encontradas no Manual de Utilização Hadron Xtorm – MU223000.

Interface do Cartão de Memória

Os cartões de memória podem ser usados para diferentes tipos de armazenamento de dados como: logs de usuários, páginas web, documentação de projeto e arquivos fontes. Mais informações sobre como usar interface de cartão de memória podem ser encontradas em Cartão de Memória.

	HX3040
Capacidade máxima	8 Gbytes
Capacidade mínima	2 Gbytes

Tipo	SD
Sistema de arquivos	FAT32
Remover cartão de forma segura	Sim, através de menu específico para essa função.

Tabela 2-9. Características da Interface com Cartão de Memória

Notas:

Capacidade máxima: A capacidade do cartão de memória deve ser igual ou inferior a este limite para o correto funcionamento na UCP Hadron Xtorm, podendo a UCP não reconhecer o cartão ou ocorrer perdas de dados durante transferências.

Capacidade mínima: A capacidade do cartão de memória deve ser igual ou superior a este limite para o seu correto funcionamento na UCP Hadron Xtorm, podendo a UCP não reconhecer o cartão ou ocorrer perdas de dados durante transferências.

Sistema de arquivos: É recomendado formatar a memória utilizando a própria UCP Hadron Xtorm, caso contrário poderá ocorrer perda de desempenho no acesso a interface do cartão de memória.

Compatibilidade com Demais Produtos

A tabela a seguir traz informações referentes à compatibilidade da série Xtorm com outros produtos:

Produto	Versão de firmware
NX1001	1.2.0.2 ou superior
NX1005	1.2.0.3 ou superior
NX2001	1.2.0.2 ou superior
NX2020	1.2.0.2 ou superior
NX6000	1.2.0.2 ou superior
NX6010	1.0.0.0 ou superior
NX6020	1.0.0.0 ou superior
NX6100	1.2.0.1 ou superior

Tabela 2-10. Compatibilidade com Demais Produtos

Desempenho

O desempenho da UCP da Série Hadron Xtorm depende:

- Tempo da Aplicação do Usuário
- Intervalo da Aplicação
- Tempo do Sistema Operacional
- Quantidade de módulos (dados de processo, entradas/saídas, entre outros)

Tempos de Aplicação

O tempo de execução de uma aplicação da UCP da Série Hadron Xtorm depende das seguintes variáveis:

- Tempo de leitura das entradas (locais e remotas)
- Tempo de execução das tarefas
- Tempo de escrita das saídas (locais e remotas)

É importante ressaltar que o tempo de execução da tarefa “MainTask” será diretamente influenciado pela tarefa de sistema “Configuration”, uma tarefa de alta prioridade, executada periodicamente pelo sistema. A tarefa “Configuration” poderá interromper a “MainTask”.

Tempos para Execução de Instruções

Na Tabela 2-11, encontram-se os tempos necessários para a realização de diferentes instruções na UCP da Série Hadron Xtorm:

Instrução	Linguagem	Variáveis	Tempos de Execução (us)
1000 Contatos	LD	BOOL	6
1000 Divisões	ST	INT	91
		REAL	91
	IL	INT	45
		REAL	87
	LD	INT	44
		REAL	86
1000 Multiplicações	ST	INT	31
		REAL	66
	IL	INT	16
		REAL	24
	LD	INT	15
		REAL	23
1000 Somas	ST	INT	29
		REAL	68
	IL	INT	16
		REAL	24
	LD	INT	15
		REAL	24

Tabela 2-11. Tempos de Instruções

Tempos de Inicialização

A UCP da Série Hadron Xtorm possui tempo de inicialização de 25 segundos, sendo que a tela inicial com o logotipo HADRON XTORM (Splash) é apresentada depois de 7 segundos da energização.

O tempo de inicialização dos módulos de E/S depende de vários fatores tais como número de módulos no barramento, intervalo da “MainTask”, etc.

Produtos Relacionados

Os seguintes produtos devem ser adquiridos separadamente quando necessário:

Código	Descrição
HD8500	MasterTool Xtorm
AL-2600	Derivador e terminador de rede RS-485
AL-2301	Cabo de rede RS-485 (até 1000 metros)
AL-2306	Cabo de rede RS-485 (até 500 metros)
AL-1729	Cabo RJ45-CMDB9
AL-1748	Cabo CMDB9-CFDB9
AL-1752	Cabo CMDB9-CMDB9
AL-1753	Cabo CMDB9-CMDB25
AL-1754	Cabo CMDB9-CFDB9
AL-1762	Cabo CMDB9- CMDB9
AL-1763	Cabo CMDB9-borneira
NX9202	Cabo RJ45-RJ45 2 m
NX9205	Cabo RJ45-RJ45 5 m
NX9210	Cabo RJ45-RJ45 10 m

NX9101	Cartão microSD de 8 GB com adaptadores SD e miniSD
HX9401	Conector com 6 terminais
HX9402	Conector com 10 terminais
HX9405	Conector com 4 terminais
HX9102	Tampa de conector de bastidor

Tabela 2-12. Produtos Relacionados

Notas:

HD8500: MasterTool Xtorm é a ferramenta de programação utilizada para a Série Hadron.

AL-2600: Este módulo é utilizado para derivação e terminação de uma rede RS-422/485. Para cada nó da rede, deve existir um AL-2600. Os módulos AL-2600 que estiverem nas extremidades da rede devem ser configurados como terminação, exceto quando há um dispositivo com terminação interna ativa, o restante deve ser configurado como derivação.

AL-2301 e AL-2306: Cabo blindado de dois pares trançados, sem conectores, para ser utilizado em redes RS-485 e redes RS-422, com comprimento máximo de 1000 e 500 metros, respectivamente.

AL-1729: Cabo padrão RS-232C com um conector RJ45 e um conector DB9 macho para comunicação entre as UCPs da Série Hadron Xtorm e outros produtos Altus das Séries DUO, Piccolo e Ponto.

AL-1748: Cabo padrão RS-232C com um conector DB9 macho e um conector DB9 fêmea para comunicação entre UCPs da Série Hadron Xtorm e outros produtos Altus da Série Cimrex.

AL-1752: Cabo padrão RS-232C com dois conectores DB9 macho para comunicação entre a UCP da Série Hadron Xtorm e outros produtos Altus da Série H e IHMs da Série IX.

AL-1753: Cabo padrão RS-232C com um conector DB9 macho e um conector DB25 macho para comunicação entre as UCPs da Série Hadron Xtorm e outros produtos Altus da Série H.

AL-1754: Cabo padrão RS-232C com um conector DB9 macho e um conector DB9 fêmea para comunicação entre as UCPs da Série Hadron Xtorm e outros produtos Altus da Série Exter ou porta Serial padrão RS-232C de um microcomputador.

AL-1762: Cabo padrão RS-232C com dois conectores DB9 macho para comunicação entre as UCPs da Série Hadron Xtorm e também para comunicação entre UCPs da Série Nexto.

AL-1763: Cabo com um conector DB9 macho e terminais para comunicação entre a UCP da Série Hadron Xtorm e produtos com bornes padrão RS-232C/RS-485/RS-422.

NX9202, NX9205 e NX9210: Cabo Ethernet CAT5, blindado, par trançado, com conectores RJ45 macho em ambas extremidades, suporta temperatura de -5 ºC a 70 ºC, para ser utilizado em redes Ethernet, com comprimento de 2, 5 e 10 metros, respectivamente.

HX9401, HX9402 e HX9405: Conectores de 6, 10 e 4 terminais, respectivamente, o conector de 6 posições, HX9401, é usado exclusivamente para a fonte HX8320. O conector de 10 posições, HX9402, pode ser usado em qualquer um dos módulos de entrada e saída da série Xtorm. O conector de 4 terminais, HX9405, é de uso exclusivo do canal IRIG-B da UCP HX3040. Suportam temperaturas de -5 ºC a 70 ºC.

HX9102: O HX9102 consiste em uma tampa protetora de conectores dos bastidores da Série Hadron Xtorm. Esta tampa foi desenvolvida para fornecer uma elevada proteção aos conectores não utilizados do bastidor. É recomendado ao usuário, que todas as posições não utilizadas sejam protegidas com o produto HX9102, suporta temperaturas de -5 °C a 70 °C.

3. Instalação

Este capítulo apresenta os procedimentos necessários para a instalação física dos produtos da Série Hadron Xtorm, bem como os cuidados que se deve ter com outras instalações existentes no armário elétrico ocupado pelo UCP.

Inspeção Visual

Antes de proceder a instalação, é recomendável fazer uma inspeção visual cuidadosa dos equipamentos, verificando se não há danos causados pelo transporte nos mesmos. Verifique se todos os componentes de seu pedido estão em perfeito estado e qualquer problema detectado deve ser informado à companhia transportadora e ao representante ou distribuidor Altus mais próximo.

CUIDADO:
Antes de retirar os módulos da embalagem, é importante a descarga de eventuais potenciais estáticos acumulados no corpo. Para isso deve-se tocar (com as mãos nuas) em uma superfície metálica aterrada qualquer, antes de manipular os módulos. Tal procedimento garante que os níveis de eletricidade estática suportados pelo módulo não serão ultrapassados.

É importante registrar o número de série de cada equipamento recebido, bem como as revisões de software, caso existentes. Essas informações serão necessárias caso necessite contatar o Suporte da Altus.

Instalação Mecânica

Fixação do Bastidor

Furação para Bastidor 9 Posições

A fixação do bastidor de 9 posições deve ser feita através de cinco parafusos M4 conforme mostrado na Figura 3-1.

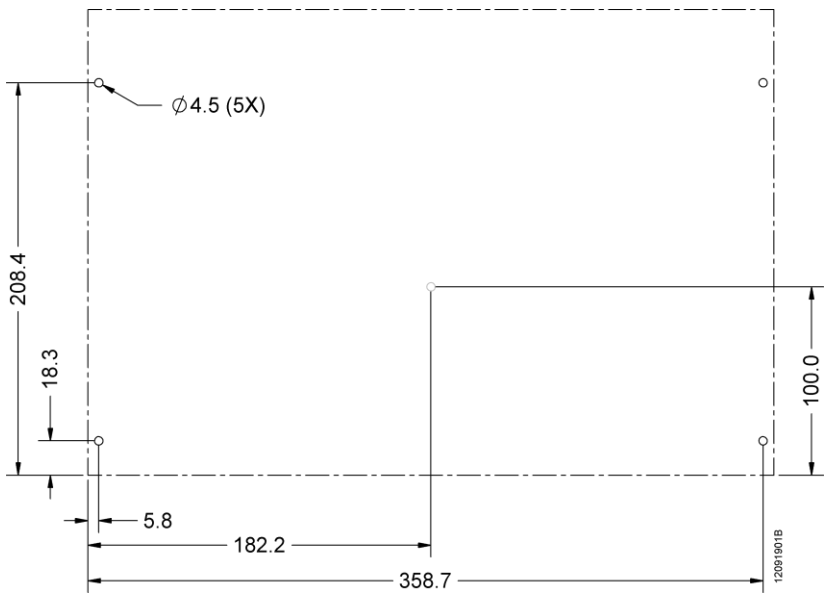


Figura 3-1. Furação para Fixação Bastidor 9 Posições

Furação para Bastidor 18 posições

A fixação do bastidor de 18 posições deve ser feita através de seis parafusos M4 conforme mostrado na Figura 3-2.

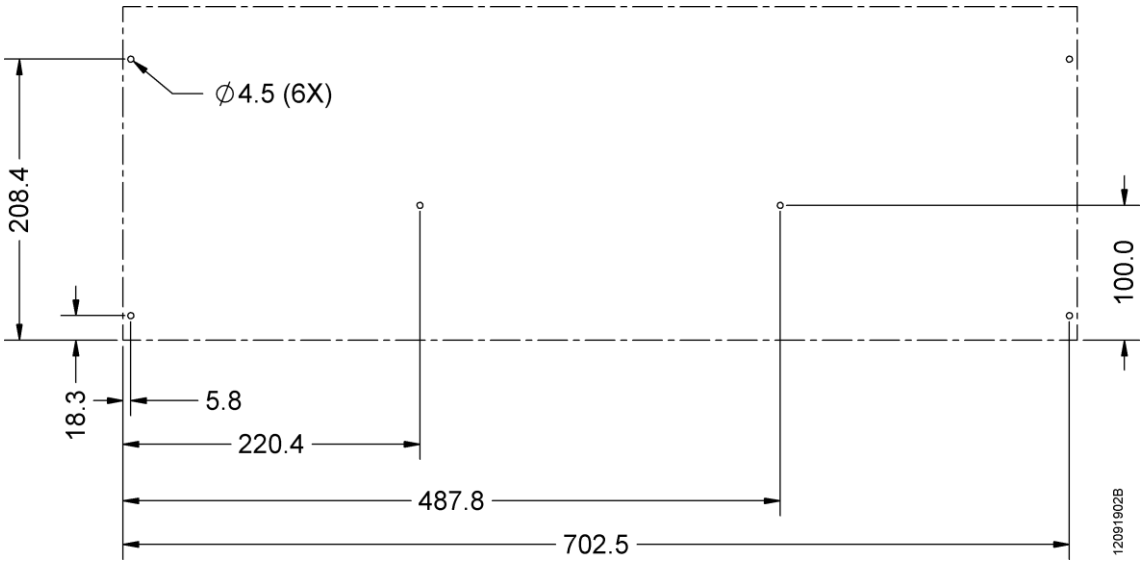


Figura 3-2. Furação para Fixação Bastidor 18 Posições

Em todos os furos devem ser utilizados parafusos tipo M4 panela fenda cruzada DIN 7985. Estes parafusos podem ser fixados diretamente no painel ou em porcas, quando a espessura do painel for insuficiente para possuir rosca. Ao utilizar porcas, recomenda-se o uso de porcas do tipo auto travante de forma a evitar que a mesma solte.

ATENÇÃO: O capítulo Projeto do Painel deve ser consultado para verificar as distâncias mínimas que devem existir entre o bastidor da Série Hadron Xtorm e os demais componentes localizados no painel elétrico.

Montagem

Antes da inserção do bastidor no painel, os parafusos dos furos tipo 1 (Figura 3-1) devem estar parcialmente inseridos.

Alinhar o bastidor aos dois parafusos tipo 1 e encostar o mesmo no fundo do painel elétrico. A Figura 3-3 indica como deve ser feito tal procedimento.

Obs.: Algumas figuras utilizadas neste item não mostram a placa de circuito impresso do bastidor por motivos de simplificar o entendimento do procedimento.

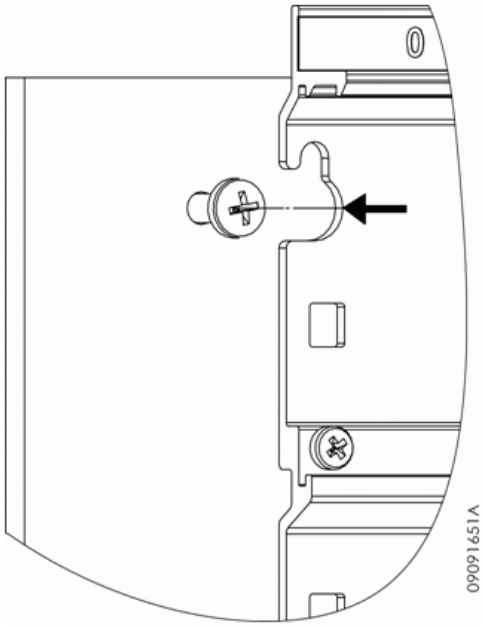


Figura 3-3. Fixação Bastidor – Alinhamento

Após, deve-se realizar um movimento de tal forma que os parafusos dos furos tipo 1 fiquem encaixados na parte menor do rasgo do bastidor, conforme mostrado na Figura 3-4.

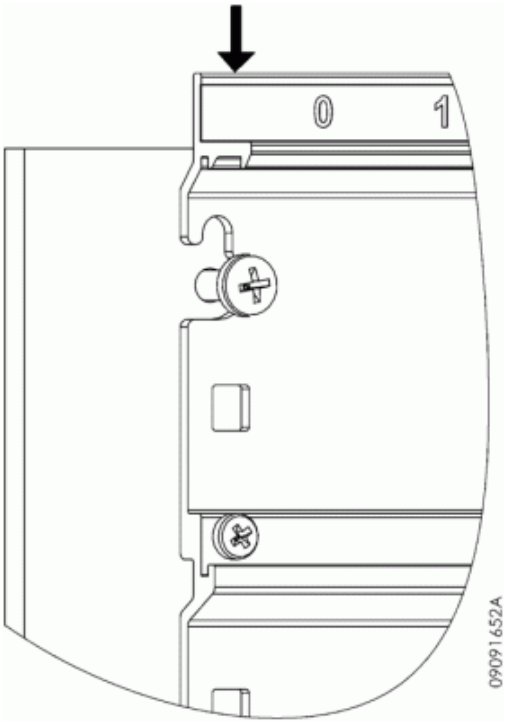


Figura 3-4. Fixação Bastidor – Encaixe nos Parafusos

Após a completa inserção do bastidor, todos os parafusos utilizados para a fixação do mesmo devem ser montados.

Remoção

Para remover o bastidor, é necessário realizar a sequência inversa indicada no item Montagem.

Inserção dos Módulos

O exemplo a seguir mostra um módulo da Série Hadron Xtorm genérico de modo que o procedimento deve ser seguido para todos os módulos da Série.

Primeiramente deve ser encaixada a parte inferior do módulo, que serve como guia para a correta inserção ao bastidor. Ao encaixar a parte inferior do módulo, deve ser verificado se os pinos guias estão corretamente encaixados aos rasgos do bastidor correspondente a uma determinada posição. A Figura 3-5 mostra como a parte do inferior do módulo deve estar posicionada em relação ao bastidor para a correta inserção. Deve ser inserido apenas um módulo por vez.

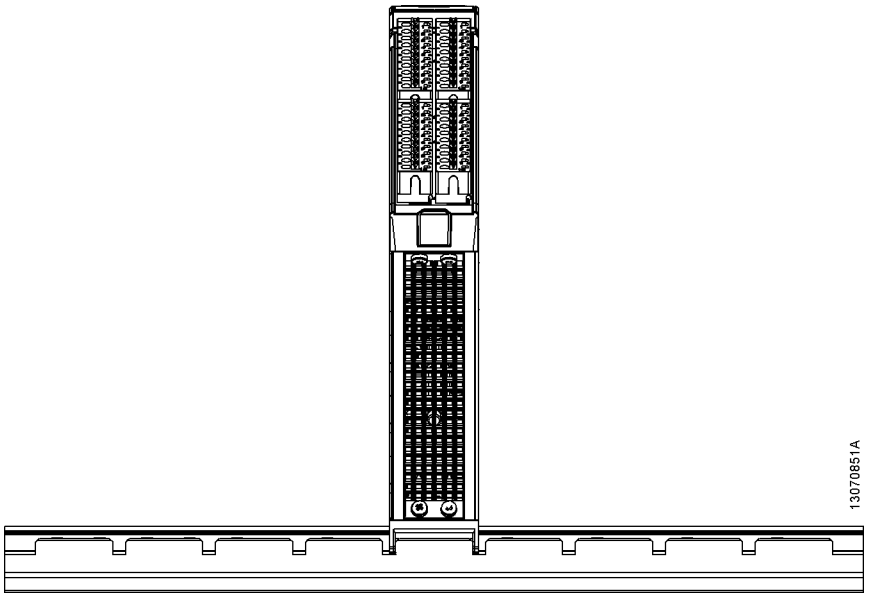


Figura 3-5. HX6000 e Bastidor

Após encaixar a parte inferior do módulo conforme descrito acima, deve-se exercer um movimento único e contínuo de rotação, através do cursor de fixação que deverá estar na posição de trava, de forma que a parte inferior do módulo fique no mesmo local e a trava de fixação encaixe na parte superior do bastidor. Após a conclusão deste procedimento, devem-se aguardar no mínimo dois segundos para a execução de um novo procedimento de inserção/remoção. A Figura 3-6 mostra o movimento que deve ser executado.

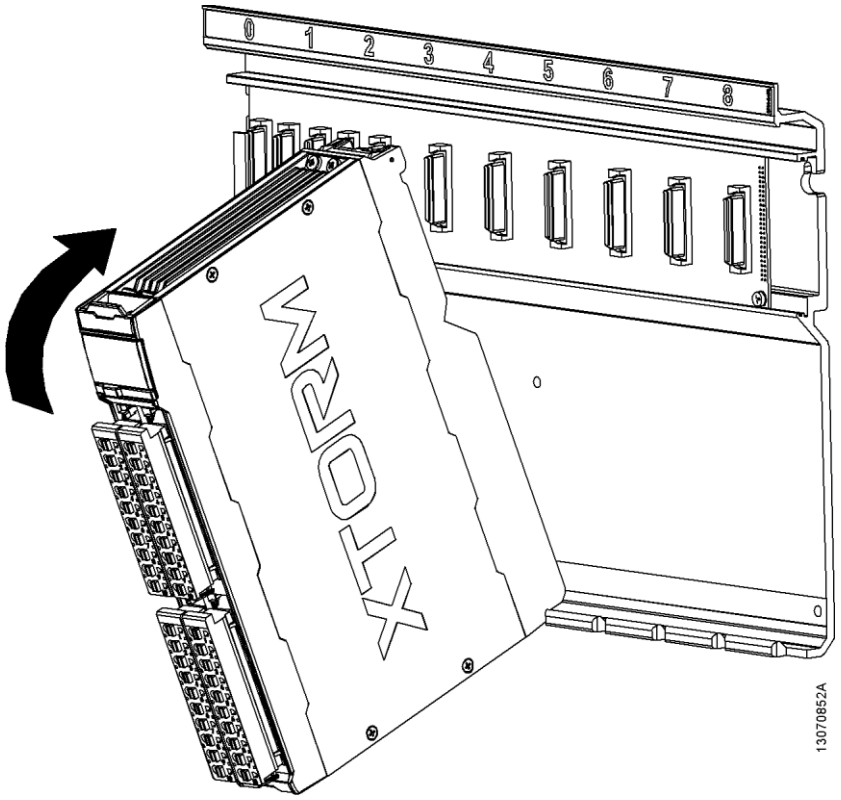


Figura 3-6. Movimento para Encaixe no Bastidor

Caso o usuário siga corretamente os procedimentos descritos acima, o módulo estará perfeitamente conectado ao barramento, conforme mostrado na Figura 3-7.

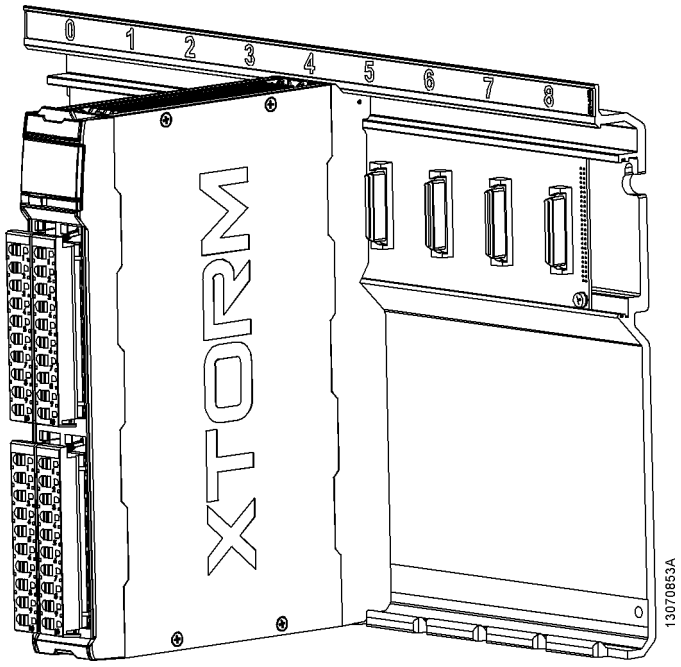


Figura 3-7. Módulo Encaixado no Bastidor

O módulo não pode ser conectado de outra forma ao bastidor. A tentativa de inserção do módulo de forma errada pode causar danos irreparáveis ao mesmo. A Figura 3-8 mostra uma maneira de como NÃO se deve conectar os módulos da Série Hadron Xtorm no bastidor.

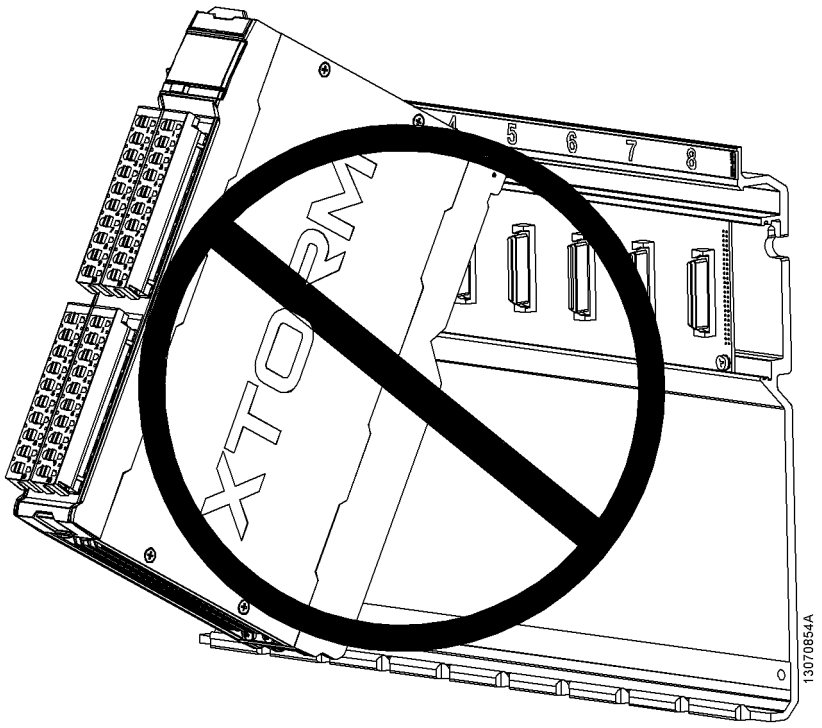


Figura 3-8. Forma Incorreta de Inserção

Remoção de Módulos

O exemplo a seguir mostra um módulo da Série Hadron Xtorm genérico de modo que o procedimento deve ser seguido para todos os módulos da Série.

Primeiramente, deve se puxar o cursor de fixação Figura 3-9 (1), a fim de destravá-lo do barramento e então rotacionar em um movimento único e contínuo, no sentido indicado na Figura 3-9 (2).

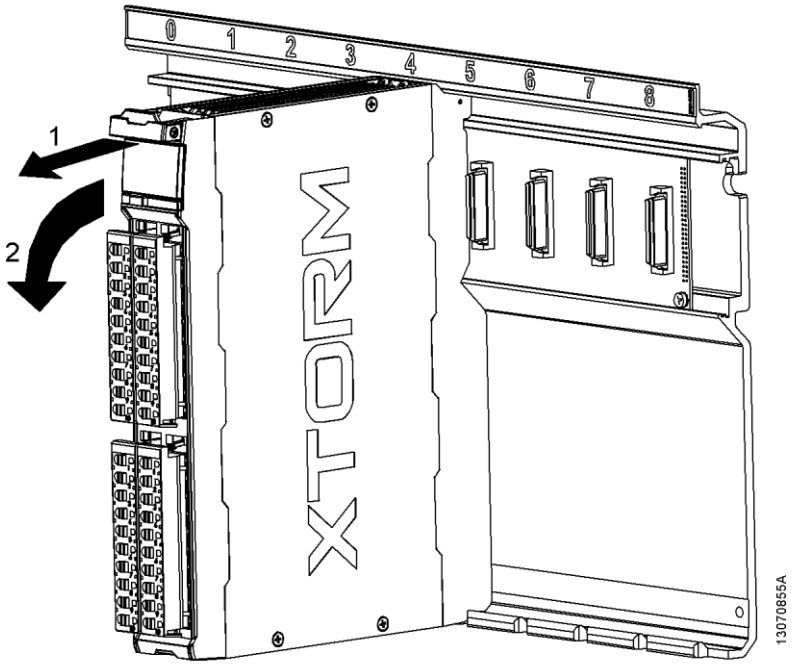


Figura 3-9. Desconexão do Barramento

Caso o usuário siga corretamente os procedimentos descritos acima, o módulo será desconectado do barramento e basta retirar o mesmo conforme indicado na Figura 3-10. Deve ser removido apenas um módulo por vez. Após a conclusão deste procedimento, devem-se aguardar no mínimo dois segundos para a execução de um novo procedimento de inserção/remoção.

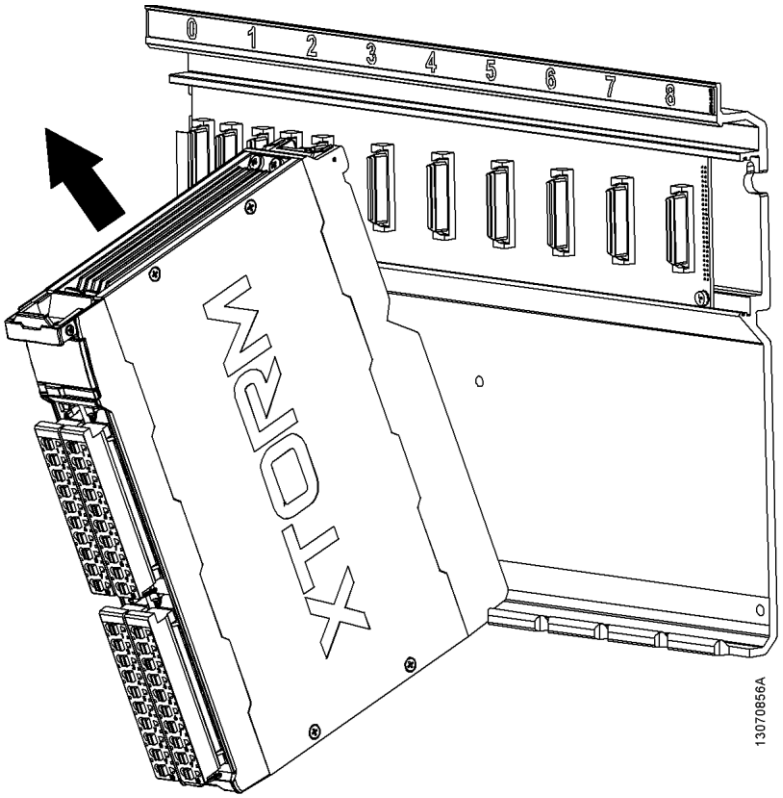


Figura 3-10. Módulo sendo Removido do Bastidor

Módulos de E/S

Os módulos de entrada e saída da Série Hadron Xtorm possuem algumas características especificas que serão abordadas neste item.

Bornes de E/S

Os bornes dos módulos de E/S da Série Xtorm utilizam o sistema de fixação de fiação com bornes tipo mola não necessitando de parafuso para tal fim.

Identificação

Todos os pinos dos bornes de E/S possuem numeração de 1 a 10. A relação entre o número do pino e a respectiva funcionalidade do mesmo para um módulo específico é definida no documento Característica Técnica do respectivo módulo.

Diagrama de Instalação

O diagrama de instalação de cada módulo é definido no documento Característica Técnica do respectivo módulo.

Tampa de Conector de Bastidor

A tampa de conector de bastidor deve ser utilizada nas posições não utilizadas do bastidor para proteção do conector contra contatos indevidos e impurezas.

Inserção da Tampa de Conector de Bastidor

A tampa de conector HX9102 da Série Hadron Xtorm, deve ser encaixada no conector iniciando-se pela parte inferior e finalizando pela parte superior, pressionando-a até encaixar completamente, conforme a Figura 3-11.

Na Figura 3-12 a tampa de conector aparece completamente encaixada.

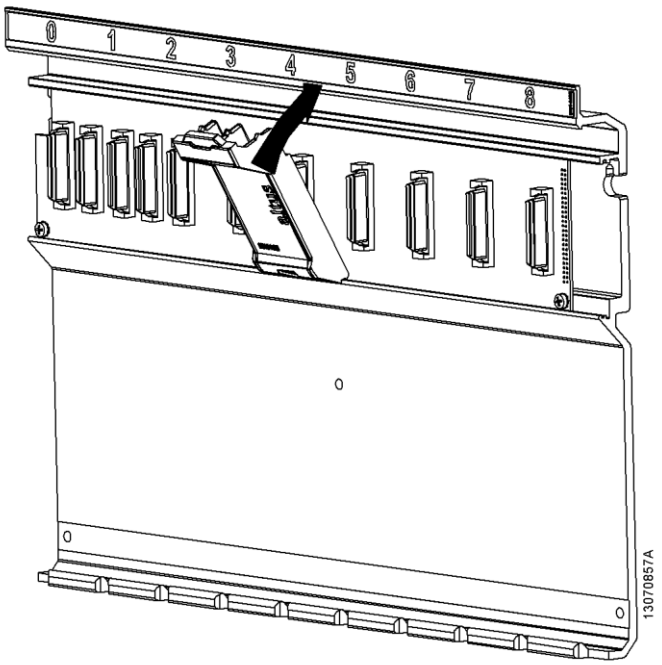


Figura 3-11. Inserção da Tampa de Conector de Bastidor

Remoção da Tampa de Conector de Bastidor

Para remover a tampa de conector HX9102 da Série Hadron Xtorm, basta puxá-la pela borda superior, conforme Figura 3-12.

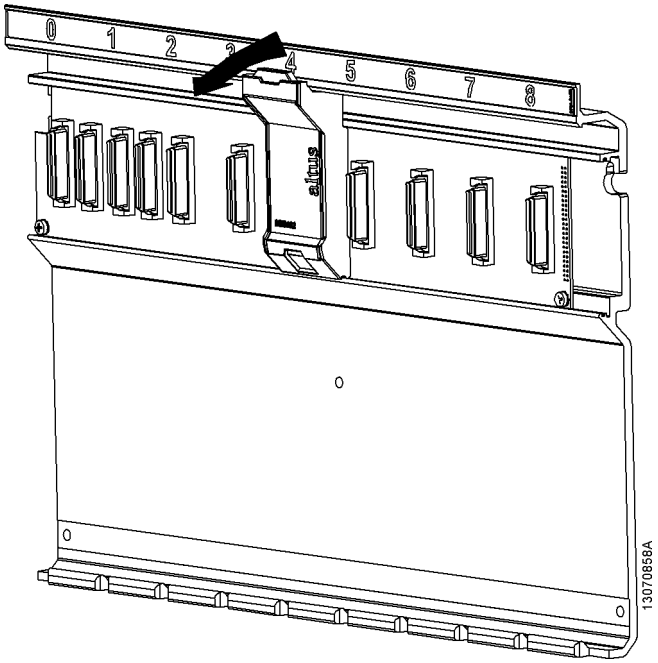


Figura 3-12. Remoção da Tampa de Conector de Bastidor

Instalação Elétrica

PERIGO:
Ao realizar qualquer instalação em um painel elétrico, certifique-se de que a alimentação geral do armário esteja DESLIGADA.

Segurança Elétrica

ATENÇÃO:
O pino 5 do borne do módulo HX8320 é o terra de proteção e obrigatoriamente deve ser ligado ao aterramento local com uma boa conexão, garantindo uma impedância máxima de 0,1 Ω.

PERIGO:
Os módulos HX8320, HX1120 e HX2320 utilizam tensões entre 91 e 135 Vdc nas interfaces de alimentação, entradas digitais e saídas digitais, respectivamente. Portanto, deve-se tomar cuidado ao manusear os cabos e bornes destes módulos durante as etapas de instalação, utilização e manutenção, verificando sempre se eles estão em bom estado de conservação, sem cortes ou rompimentos. Para garantir maior segurança, os bornes dos módulos tem classificação IP20.

ATENÇÃO:
A interface serial COM 1 (RS-232/RS-485) da UCP HX3040 possui um conector DB9 que tem classificação IP20, e, portanto exime o operador de risco de choque elétrico ao tocar neste. Entretanto, deve-se atentar para não tocar nos pinos do cabo que estiver ligado à esta interface. E ainda, em função desta ser perigosa em caso de falha simples, ela deve ser ligada apenas a produtos que contenham isolação superior a 820 Vac desta interface para outras.

ATENÇÃO:
As Placas de Circuito Impresso dos bastidores HX9001 e HX9003, que podem, em condições de falha simples, apresentar tensões perigosas estão representadas nas regiões destacadas em verde na Figura 3-13. Portanto, estas regiões devem ser tocadas apenas com o painel desenergizado.

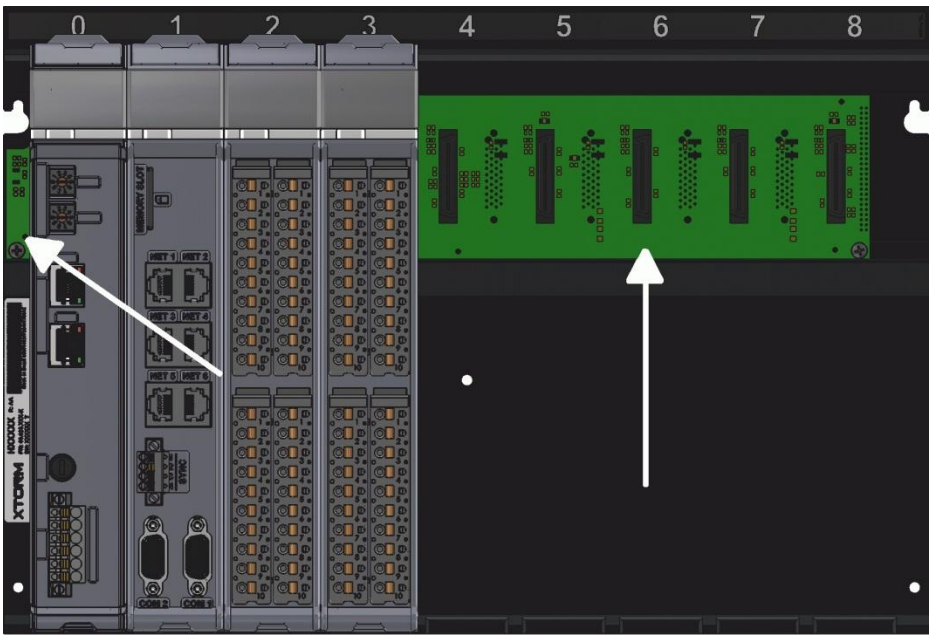


Figura 3-13. Região da Placas de Circuito Impresso (HX9001 e HX9003)

Bornes com Mola

Este tipo de borne possui um sistema de fixação baseado em uma mola, de elevada confiabilidade, mesmo em ambientes sujeitos a vibração (Figura 3-14 e Figura 3-15). Para sua montagem recomenda-se utilizar uma chave de fenda com 3,5 mm de largura e cabo isolado (Figura 3-15). A vantagem de seu emprego é a facilidade e rapidez de montagem dos cabos elétricos.

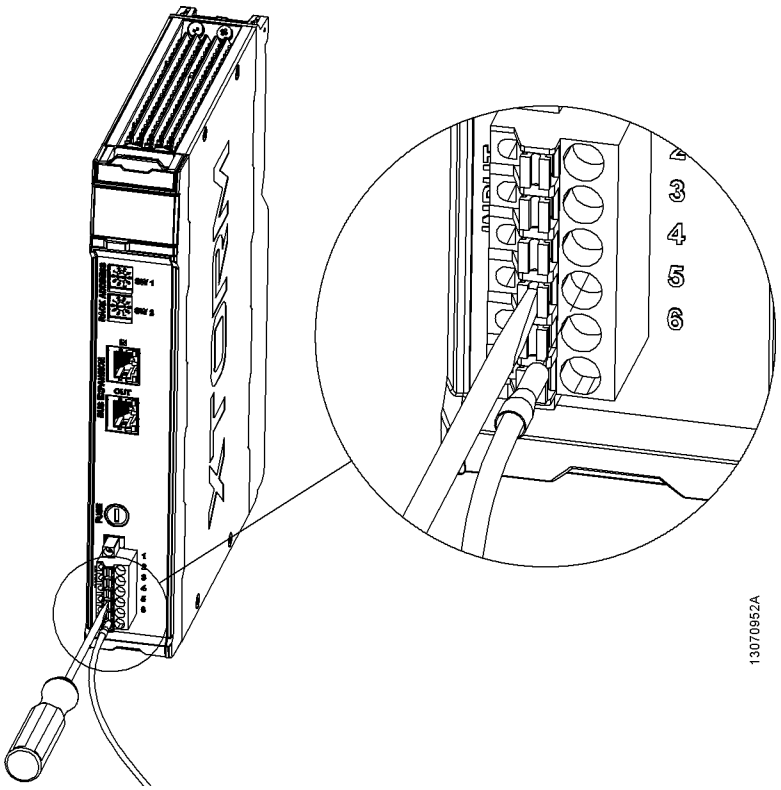


Figura 3-14. Borne Mola Módulo Fonte

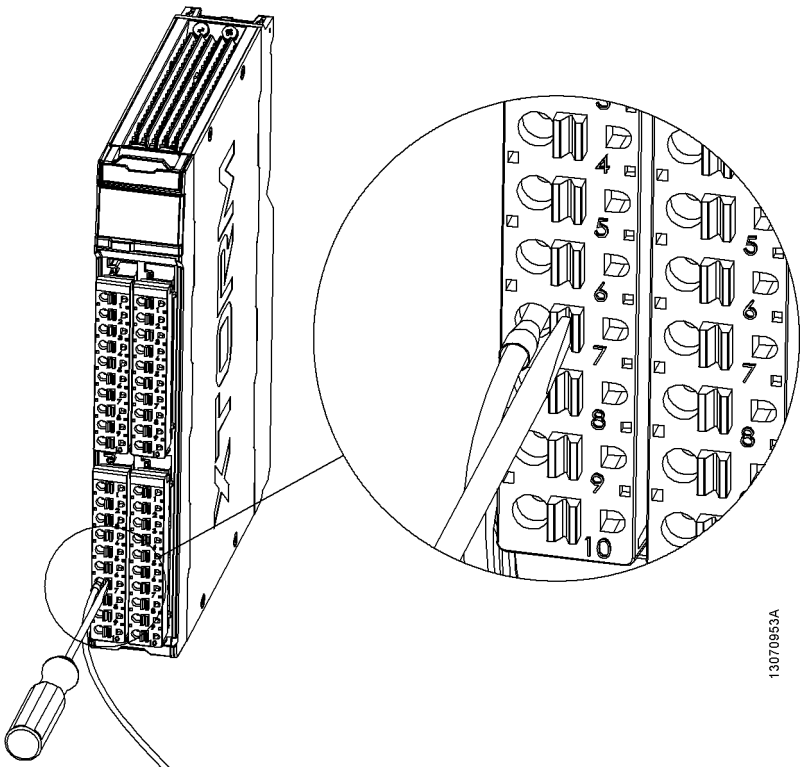


Figura 3-15. Borne Mola Módulo IO

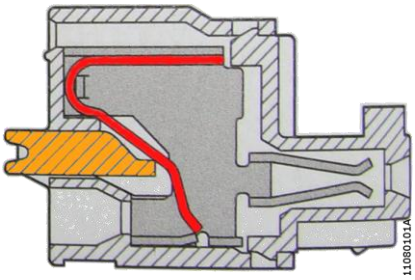


Figura 3-16. Borne Mola

Para montar o fio no borne:

- Inserir a chave de fenda no acionador do terminal para abrir a mola do borne
- Inserir o terminal do fio no borne
- Retirar a chave para fechar o borne

Inserção da Fiação

De forma a obter um comprimento semelhante de todos os fios conectados a um mesmo borne de E/S, indica-se seguir a seguinte definição referente à diferença de comprimento que cada fio subsequente ao outro deve ter antes da instalação. É importante ressaltar que o maior fio devera ser sempre o fio conectado ao pino 1 do borne de E/S.

Borne de 4 Vias

Recomenda-se utilizar fio de 0,5 mm². Crimpar terminais especificados para o para fio escolhido (0,5 mm²) em cada via.

ATENÇÃO: Utilizar terminais com comprimento A = 8 mm para garantir o efetivo contato (ver Figura 3-18).

Borne de 6 Vias

Recomenda-se utilizar fio de 2,5 mm². Crimpar os terminais especificados para o fio escolhido (2,5 mm²) em cada via.

ATENÇÃO: Utilizar terminais com comprimento A = 12 mm para garantir o efetivo contato (ver Figura 3-18).

Borne de 10 Vias

Recomenda-se utilizar fio de 0,5 a 1,5 mm². Cortar cada fio com uma diferença de 8 mm, conforme indicado na Figura 3-17. Crimpar terminais especificados para o para fio escolhido (0,5 a 1,5 mm²) em cada via.

ATENÇÃO: Utilizar terminais com comprimento A = 12 mm para garantir o efetivo contato (ver Figura 3-18).

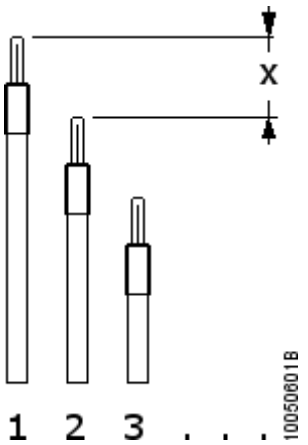


Figura 3-17. Corte fiação

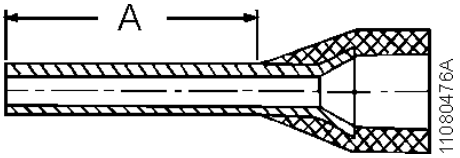


Figura 3-18. Terminal

Montagem da Fiação

Inserir os terminais no borne iniciando pelo pino 10, para o borne de 10 vias. Certifique-se de que os terminais estão completamente inseridos no borne e devidamente conectados.

Fixação da Fiação

Girar os cabos para que fiquem acomodados e fixar com amarrilha plástica conforme Figura 3-19.

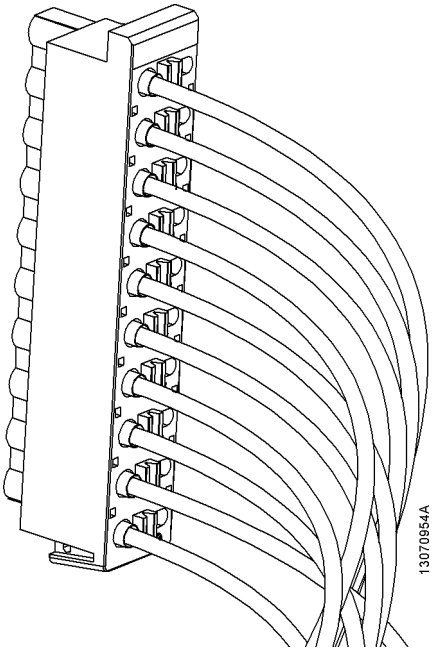


Figura 3-19. Montagem cabo no conector de 10 vias

Remoção da Fiação

Para remover os fios do borne recomenda-se utilizar uma chave de fenda com 3,5 mm de largura e cabo isolado. Inserir a chave de fenda no acionador ao lado do fio enquanto puxa o fio para fora (ver Figura 3-14 e Figura 3-15).

Conexões

A correta fixação dos cabos das UCPs e dos módulos do sistema garante a segurança do equipamento e seu correto funcionamento. Para isso, devem ser verificados os seguintes pontos:

- Os cabos junto aos bornes de ligação do painel de montagem devem estar com conexão segura e firme
- Os bornes de alimentação e aterramento das partes do sistema devem estar firmes e bem conectados, assegurando boa passagem de corrente
- A conexão do terra dos equipamentos ao terra do painel de montagem deve estar firme e com a bitola de cabo correta, para garantir bom aterramento e imunidade a ruído
- O encaixe entre os conectores e os módulos deve ser verificado de forma a garantir que o mesmo tenha sido inserido completamente, garantido uma correta conexão

Alimentações

Conferir se as tensões das alimentações estão dentro dos valores especificados nas características técnicas.

ATENÇÃO: Onde houver alta tensão, colocar etiqueta de aviso e proteções que não permitam o fácil acesso.

Fusíveis

Recomenda-se verificar os fusíveis do sistema, certificando-se que os mesmos estejam em bom estado e com valor e tipo correto, antes de energizar o sistema.

PERIGO: **Nunca se deve substituir um fusível por outro de maior valor de corrente, sob pena de causar sérios danos ao equipamento.**

Instalação Elétrica UCP

PERIGO: **Ao realizar qualquer instalação em um painel elétrico, certifique-se de que a alimentação geral do armário esteja DESLIGADA.**

A alimentação da UCP da Série Hadron Xtorm é proveniente do Módulo Fonte de Alimentação, o qual fornece tensão às UCPs através da conexão ao bastidor, não necessitando de conexões externas. O aterramento do módulo é realizado através do contato entre a mola de aterramento do módulo e o bastidor.

A Figura 3-20 ilustra o diagrama elétrico da UCP da Série Hadron Xtorm instalada em um bastidor da Série Hadron Xtorm. A mesma imagem pode ser visualizada na lateral esquerda da mecânica da UCP.

A disposição dos conectores e bornes na figura é meramente ilustrativa.

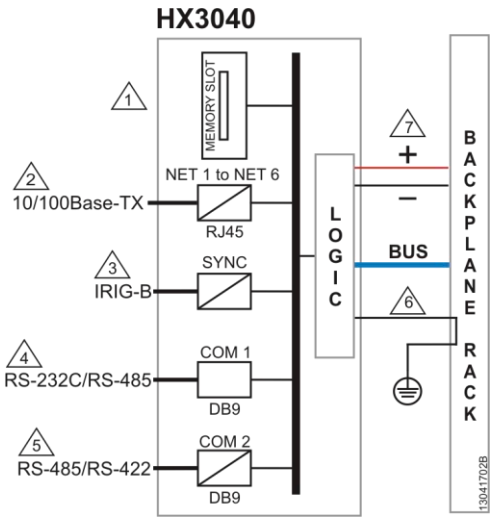


Figura 3-20. Diagrama Elétrico da UCP da Série Hadron Xtorm

Notas do Diagrama:

1. Interface para cartão de memória
2. Interface Ethernet padrão 10/100Base-TX para programação, depuração e conexão aos protocolos suportados
3. Interface IRIG-B padrão para conexão ao sinal de sincronismo de tempo. Recomenda-se a utilização de cabos de seção de 1,5mm²
4. Interface serial padrão RS-232C/RS-485 para conexão à rede MODBUS RTU ou outros protocolos
5. Interface serial padrão RS-485/RS-422 para conexão à rede MODBUS RTU ou outros protocolos. A escolha do tipo de interface física depende do cabo utilizado
6. O módulo é aterrado através dos bastidores da Série Hadron Xtorm
7. A alimentação do módulo é proveniente da conexão ao bastidor, não necessitando de conexões externas

Conexão com a Rede Ethernet

As interfaces isoladas de comunicação NET 1 ao NET 6 possibilitam a conexão com uma rede Ethernet, onde a interface NET 1 (ou NET 2 quando estiver configurada em par NIC Teaming com a NET 1) deve ser usada na comunicação com o MasterTool Xtorm.

A conexão com a rede Ethernet utiliza cabos tipo par trançado (10/100Base-TX), sendo que a detecção da velocidade é realizada automaticamente pela UCP Hadron Xtorm. Este cabo deve ter uma de suas extremidades ligadas à interface que se pretende utilizar e a outra ao HUB, switch, microcomputador ou outro ponto de rede Ethernet.

Endereço IP

A interface Ethernet NET 1 é utilizada para comunicação Ethernet e para configurar a UCP. Para que isso seja possível, esta vem configurada de fábrica com os seguintes parâmetros:

	NET 1
Endereço IP	192.168.15.1
Máscara de Subrede	255.255.255.0
Endereço do Gateway	192.168.15.253

Tabela 3-1. Parâmetros de Fábrica da Interface Ethernet NET 1

Os parâmetros Endereço IP e Máscara de Subrede podem ser visualizados no visor gráfico da UCP através do menu de parâmetros, conforme descrito no capítulo Menu Informativo e de Configuração da UCP.

Inicialmente, deve-se conectar a interface NET 1 da UCP a uma rede ou microcomputador com a mesma subrede para comunicação com o MasterTool Xtorm, onde os parâmetros de rede podem ser alterados. Para maiores detalhes sobre configuração e alteração de parâmetros de rede, verifique o capítulo Configuração das Interfaces Ethernet.

A interface de Ethernet NET 2 é utilizada para comunicação Ethernet e vem configurada de fábrica com os seguintes parâmetros:

	NET 2
Endereço IP	192.168.16.1
Máscara de Subrede	255.255.255.0
Endereço do Gateway	192.168.16.253

Tabela 3-2. Parâmetros de Fábrica da Interface Ethernet NET 2

A interface de Ethernet NET 3 é utilizada para comunicação Ethernet e vem configurada de fábrica com os seguintes parâmetros:

	NET 3
Endereço IP	192.168.17.1
Máscara de Subrede	255.255.255.0
Endereço do Gateway	192.168.17.253

Tabela 3-3. Parâmetros de Fábrica da Interface Ethernet NET 3

A interface de Ethernet NET 4 é utilizada para comunicação Ethernet e vem configurada de fábrica com os seguintes parâmetros:

	NET 4
Endereço IP	192.168.18.1
Máscara de Subrede	255.255.255.0
Endereço do Gateway	192.168.18.253

Tabela 3-4. Parâmetros de Fábrica da Interface Ethernet NET 4

A interface de Ethernet NET 5 é utilizada para comunicação Ethernet e vem configurada de fábrica com os seguintes parâmetros:

	NET 5
Endereço IP	192.168.19.1
Máscara de Subrede	255.255.255.0
Endereço do Gateway	192.168.19.253

Tabela 3-5. Parâmetros de Fábrica da Interface Ethernet NET 5

A interface de Ethernet NET 6 é utilizada para comunicação Ethernet e vem configurada de fábrica com os seguintes parâmetros:

	NET 6
Endereço IP	192.168.20.1
Máscara de Subrede	255.255.255.0
Endereço do Gateway	192.168.20.253

Tabela 3-6. Parâmetros de Fábrica da Interface Ethernet NET 6

Os parâmetros de rede das interfaces NET 1 ao NET 6 podem ser alterados através do MasterTool Xtorm. Para maiores detalhes sobre configuração e alteração de parâmetros de rede, verifique o capítulo Configuração das Interfaces Ethernet.

ARP Gratuito

A interface de Ethernet NETx envia espontaneamente pacotes do tipo ARP, em broadcast, informando seu endereço de IP e MAC para todos os dispositivos interligados à rede. Estes pacotes são enviados durante o download de uma nova aplicação pelo software MasterTool Xtorm e na inicialização da UCP, quando a aplicação entra em modo Run.

São disparados 5 comandos ARP com um intervalo inicial de 200 ms, dobrando o intervalo entre cada novo comando disparado, totalizando 3 s. Ex.: Primeiro disparo ocorre no tempo 0, o segundo disparo no tempo 200 ms, o terceiro disparo no tempo 600 ms e assim até o quinto disparo no tempo 3 s.

Instalação do Cabo de Rede

As portas Ethernet da UCP da Série Hadron Xtorm, identificadas no painel por NET 1 ao NET 6, possuem pinagem padrão, sendo a mesma utilizada, por exemplo, em computadores pessoais. O tipo de conector, tipo de cabo, nível físico, entre outros detalhes para interligar a UCP ao dispositivo de acesso à rede Ethernet, são definidos no capítulo Descrição Técnica - Interfaces Ethernet.

A Figura 3-21 e a Tabela 3-7 apresentam o conector RJ45 fêmea da UCP Hadron Xtorm, com a identificação e a descrição da pinagem válida para os níveis físicos tipo 10Base-T e 100Base-TX.

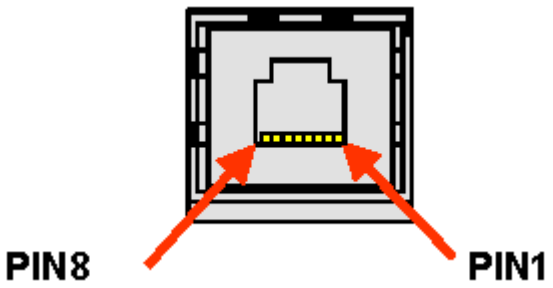


Figura 3-21 Conector RJ45 fêmea da UCP Hadron Xtorm

Pino	Sinal	Descrição
1	TXD+	Transmissão de dados, positivo
2	TXD-	Transmissão de dados, negativo
3	RXD+	Recepção de dados, positivo
4	-	Não utilizado
5	-	Não utilizado
6	RXD-	Recepção de dados, negativo
7	-	Não utilizado
8	-	Não utilizado

Tabela 3-7. Pinagem do Conector RJ45 fêmea da UCP Hadron Xtorm

A interface pode ser conectada em uma rede de comunicação através de um hub ou switch, ou então diretamente ao equipamento com o qual irá se comunicar. Neste último caso, devido a UCP Hadron Xtorm possuir a característica de Auto Crossover, não se faz necessária a utilização de um cabo de rede denominado Cross-over, o mesmo utilizado para conectar dois computadores pessoais, ponto a ponto, através da porta Ethernet.

É importante ressaltar que entende-se por cabo de rede, um par de conectores RJ45 machos interligados entre si por um cabo UTP ou ScTP, de categoria 5, sob a configuração direta ou Cross-over. O mesmo serve para interligar dois dispositivos com porta Ethernet.

Normalmente estes cabos possuem uma trava de conexão que garante uma perfeita conexão entre o conector fêmea da interface e o conector macho do cabo. No momento da instalação, o conector macho do cabo deve ser inserido na fêmea do módulo até que se ouça um som característico ("click"), garantindo a atuação da trava. Para desconectar os mesmos deve-se utilizar a alavanca presente no conector macho.

Conexão com a Rede Serial (COM 1)

ATENÇÃO:
Para uma instalação adequada e com segurança da Rede Serial COM 1, deve-se consultar primeiramente o subcapítulo de Segurança Elétrica, parte integrante deste mesmo capítulo.

A interface não isolada de comunicação COM 1 possibilita a conexão com uma rede RS-232C ou RS-485. A seguir é apresentado o conector DB9 fêmea da UCP Hadron Xtorm, com a identificação e a descrição dos sinais.

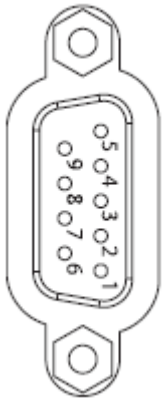


Figura 3-22. Conector DB9 fêmea da UCP Hadron Xtorm

Pino	Sinal	Interface	Descrição
1	DCD	RS-232C	Data Carrier Detect
2	TXD	RS-232C	Transmissão de dados
3	RXD	RS-232C	Recepção de dados
4	D-	RS-485	Transmissão/Recepção de dados, negativo
5	GND	RS-232C RS-485	RS-232C: Ground RS-485: Referência negativa para terminação externa
6	D+	RS-485	Transmissão/Recepção de dados, positivo
7	CTS	RS-232C	Clear to Send
8	RTS	RS-232C	Request to Send
9	+5V	RS-485	Referência positiva para terminação externa

Tabela 3-8. Pinagem do Conector DB9 fêmea da UCP Hadron Xtorm

Comunicação RS-232C

Para conexão com um dispositivo RS-232C, utilizar o cabo apropriado conforme o capítulo Descrição Técnica - Produtos Relacionados.

Comunicação RS-485 sem Terminação

Para conexão a uma rede RS-485 sem terminação na interface COM 1, deve-se conectar os terminais identificados do cabo AL-1763 nos respectivos bornes do dispositivo, conforme a Tabela 3-9.

Terminais do AL-1763	Sinais da UCP
0	Blindagem
1	Não conectado
2	Não conectado
3	D-
4	Não conectado

5	D+
6	Não conectado
7	Não conectado
8	Não conectado

Tabela 3-9. Conexão da COM 1 com RS-485 sem Terminação

O diagrama da Figura 3-23 indica como deve ser realizada a conexão dos terminais do AL-1763 na rede RS-485.

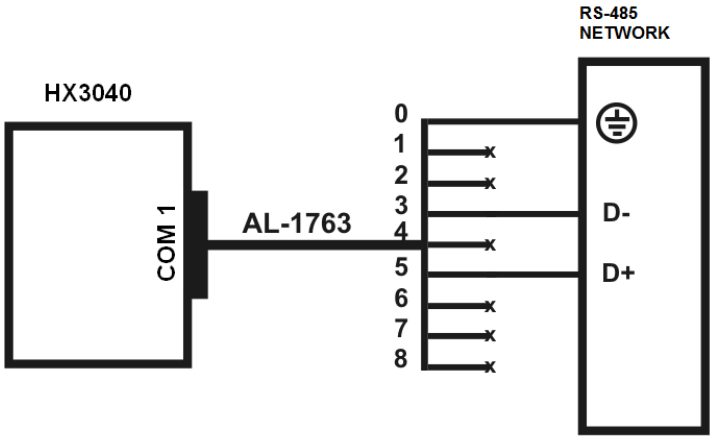


Figura 3-23. Diagrama de Conexão da COM 1 com RS-485 sem Terminação

Nota do Diagrama:

Os terminais não conectados devem ser isolados para que não haja contato entre os mesmos.

Comunicação RS-485 com Terminação Interna

A interface COM 1 não permite terminação interna. Para conexão a uma rede RS-485, utilizando a terminação interna, deve ser utilizada a interface COM 2.

Comunicação RS-485 com Terminação Externa

Para conexão a uma rede RS-485, utilizando a terminação externa da interface COM 1, deve-se conectar os terminais identificados do cabo AL-1763 nos respectivos bornes do dispositivo, conforme a Tabela 3-10.

Terminais do AL-1763	Sinais da UCP
0	Blindagem
1	Não conectado
2	Não conectado
3	D-
4	0 V
5	D+
6	Não conectado
7	Não conectado
8	+ 5 V

Tabela 3-10. Conexão da COM 1 com RS-485 com Terminação Externa

O diagrama da Figura 3-24 indica como deve ser realizada a conexão dos terminais do AL-1763 na rede RS-485.

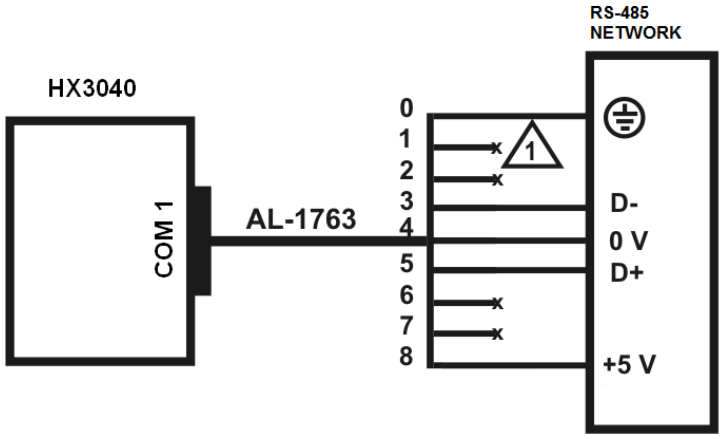


Figura 3-24. Diagrama de Conexão da COM 1 com RS-485 com Terminação Externa

Nota do Diagrama:

Os terminais não conectados devem ser isolados para que não haja contato entre os mesmos.

Conexão com a Rede Serial (COM 2)

A interface isolada de comunicação COM 2 possibilita a conexão com uma rede RS-485/422. A seguir é apresentado o conector DB9 fêmea da UCP Hadron Xtorm, com a identificação e a descrição dos sinais.

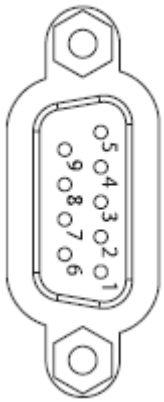


Figura 3-25 Conector DB9 fêmea da UCP Hadron Xtorm

Pino	Sinal	Descrição
1	-	Não utilizado
2	Term+	Terminação interna, positivo
3	TXD+	Transmissão de dados, positivo
4	RXD+	Recepção de dados, positivo
5	GND	Referência negativa para terminação externa
6	+5V	Referência positiva para terminação externa
7	Term-	Terminação interna, negativo
8	TXD-	Transmissão de dados, negativo
9	RXD-	Recepção de dados, negativo

Tabela 3-11. Pinagem do Conector DB9 fêmea da UCP Hadron Xtorm

Comunicação RS-485 sem Terminação

Para conexão a uma rede RS-485 sem terminação na interface COM 2, deve-se conectar os terminais identificados do cabo AL-1763 nos respectivos bornes do dispositivo, conforme a Tabela 3-12.

Terminais do AL-1763	Sinais da UCP
0	Blindagem
1	Não conectado
2	D+
3	D+
4	Não conectado
5	Não conectado
6	Não conectado
7	D-
8	D-

Tabela 3-12. Conexão da COM 2 com RS-485 sem Terminação

O diagrama da Figura 3-26 indica como deve ser realizada a conexão dos terminais do AL-1763 na borneira do dispositivo.

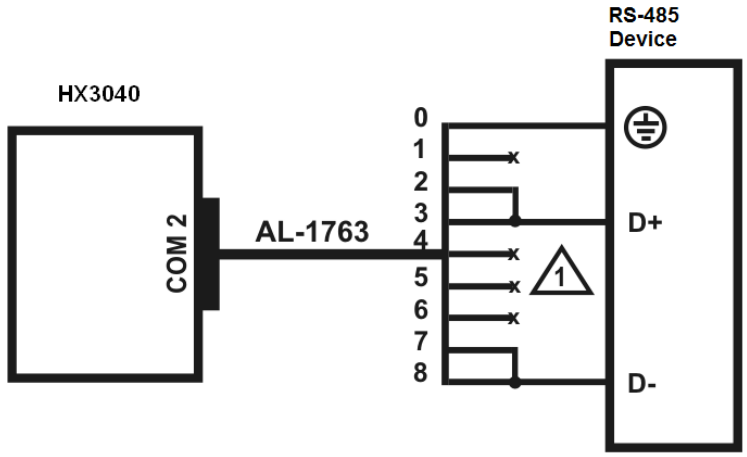


Figura 3-26. Diagrama de Conexão da COM 2 com RS-485 sem Terminação

Nota do Diagrama:

Os terminais não conectados devem ser isolados para que não haja contato entre os mesmos.

Comunicação RS-485 com Terminação Interna

Para conexão a uma rede RS-485, utilizando a terminação interna da interface COM 2, deve-se conectar os terminais identificados do cabo AL-1763 nos respectivos bornes do dispositivo, conforme a Tabela 3-13.

Terminais do AL-1763	Sinais da UCP
0	Blindagem
1	D+
2	D+
3	D+
4	Não conectado
5	Não conectado
6	D-
7	D-
8	D-

Tabela 3-13. Conexão da COM 2 com RS-485 com Terminação Interna

Obs.: A terminação interna disponível na COM 2 é do tipo estado seguro em modo aberto.

O diagrama da Figura 3-27 indica como deve ser realizada a conexão dos terminais do AL-1763 na borneira do dispositivo.

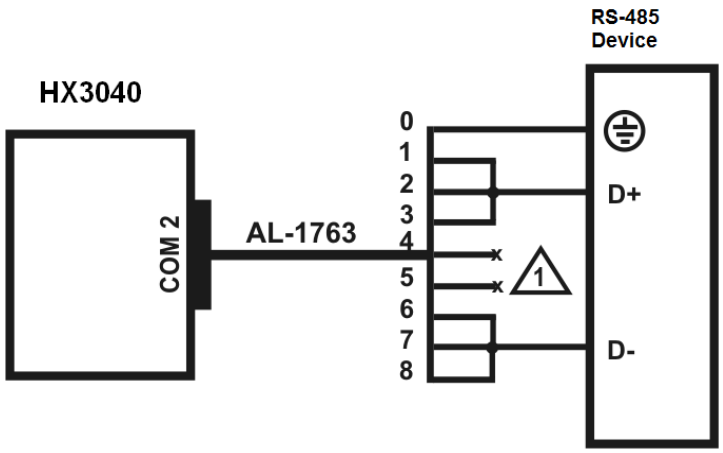


Figura 3-27. Diagrama de Conexão da COM 2 com RS-485 com Terminação Interna

Nota do Diagrama:

Os terminais não conectados devem ser isolados para que não haja contato entre os mesmos.

Comunicação RS-485 com Terminação Externa

Para conexão a uma rede RS-485, utilizando a terminação externa da interface COM 2, deve-se conectar os terminais identificados do cabo AL-1763 nos respectivos bornes do dispositivo, conforme a Tabela 3-14.

Terminais do AL-1763	Sinais da UCP
0	Blindagem
1	Não conectado
2	D+
3	D+
4	0 V
5	+5 V
6	Não conectado
7	D-
8	D-

Tabela 3-14. Conexão da COM 2 com RS-485 com Terminação Externa

O diagrama da Figura 3-28 indica como deve ser realizada a conexão dos terminais do AL-1763 na borneira do dispositivo.

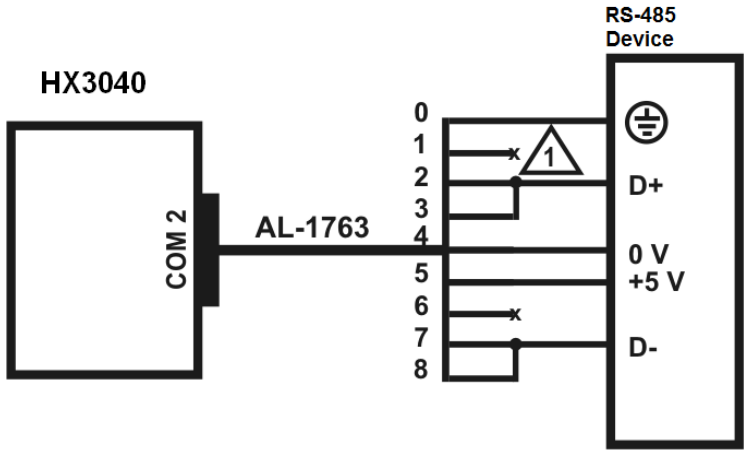


Figura 3-28. Diagrama de Conexão da COM 2 com RS-485 com Terminação Externa

Nota do Diagrama:

Os terminais não conectados devem ser isolados para que não haja contato entre os mesmos.

Comunicação RS-422 sem Terminação

Para conexão a uma rede RS-422 sem terminação na interface COM 2, deve-se conectar os terminais identificados do cabo AL-1763 nos respectivos bornes do dispositivo, conforme a Tabela 3-15.

Terminais do AL-1763	Sinais da UCP
0	Blindagem
1	Não conectado
2	TX+
3	RX+
4	Não conectado
5	Não conectado
6	Não conectado
7	TX-
8	RX-

Tabela 3-15. Conexão da COM 2 com RS-422 sem Terminação

O diagrama da Figura 3-29 indica como deve ser realizada a conexão dos terminais do AL-1763 na borneira do dispositivo.

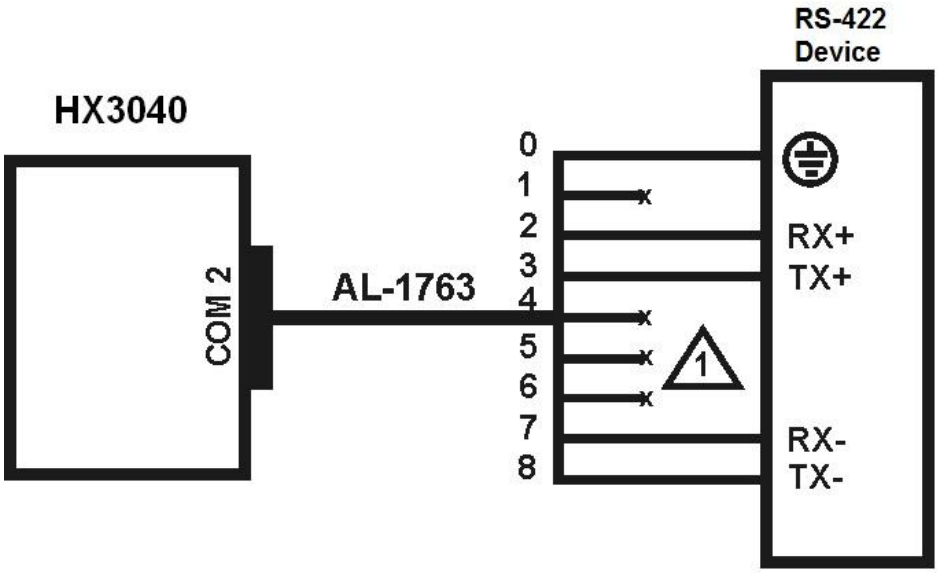


Figura 3-29. Diagrama de Conexão da COM 2 com RS-422 sem Terminação

Nota do Diagrama:

Os terminais não conectados devem ser isolados para que não haja contato entre os mesmos.

Comunicação RS-422 com Terminação Interna

Para conexão a uma rede RS-422, utilizando a terminação interna da interface COM 2, deve-se conectar os terminais identificados do cabo AL-1763 nos respectivos bornes do dispositivo, conforme a Tabela 3-16.

Terminais do AL-1763	Sinais da UCP
0	Blindagem
1	TERM+
2	TX+
3	RX+
4	Não conectado
5	Não conectado
6	TERM-
7	TX-
8	RX-

Tabela 3-16. Conexão da COM 2 com RS-422 com Terminação

Obs.: A terminação interna disponível na COM 2 é do tipo estado seguro em modo aberto.

O diagrama da Figura 3-30 indica como deve ser realizada a conexão dos terminais do AL-1763 na borneira do dispositivo.

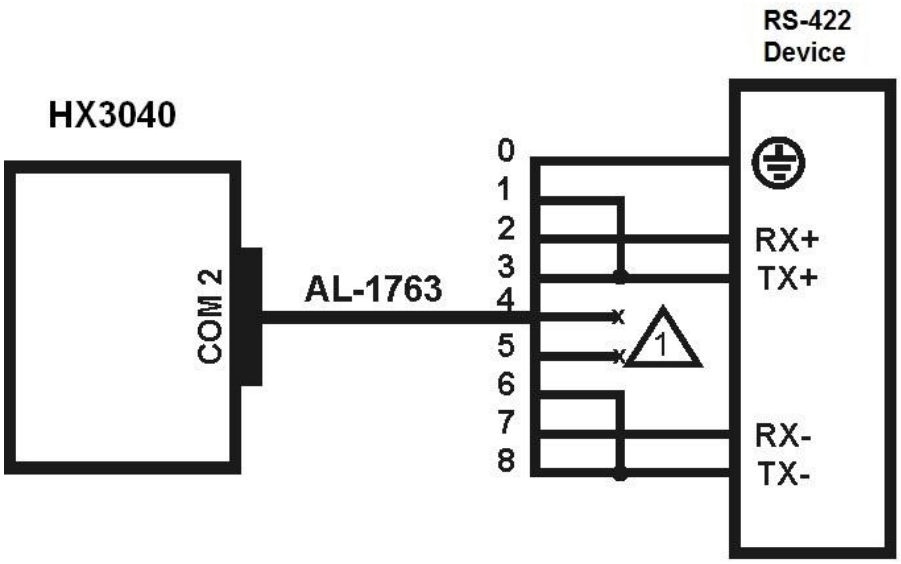


Figura 3-30. Diagrama de Conexão da COM 2 com RS-422 com Terminação Interna

Nota do Diagrama:

Os terminais não conectados devem ser isolados para que não haja contato entre os mesmos.

Comunicação RS-422 com Terminação Externa

Para conexão a uma rede RS-422, utilizando a terminação externa da interface COM 2, deve-se conectar os terminais identificados do cabo AL-1763 nos respectivos bornes do dispositivo, conforme a Tabela 3-17.

Terminais do AL-1763	Sinais da UCP
0	Blindagem
1	Não conectado
2	TX+
3	RX+
4	0 V
5	+5 V
6	Não Conectado
7	TX-
8	RX-

Tabela 3-17. Conexão da COM 2 com RS-422 com Terminação Externa

O diagrama da Figura 3-31 indica como deve ser realizada a conexão dos terminais do AL-1763 na borneira do dispositivo.

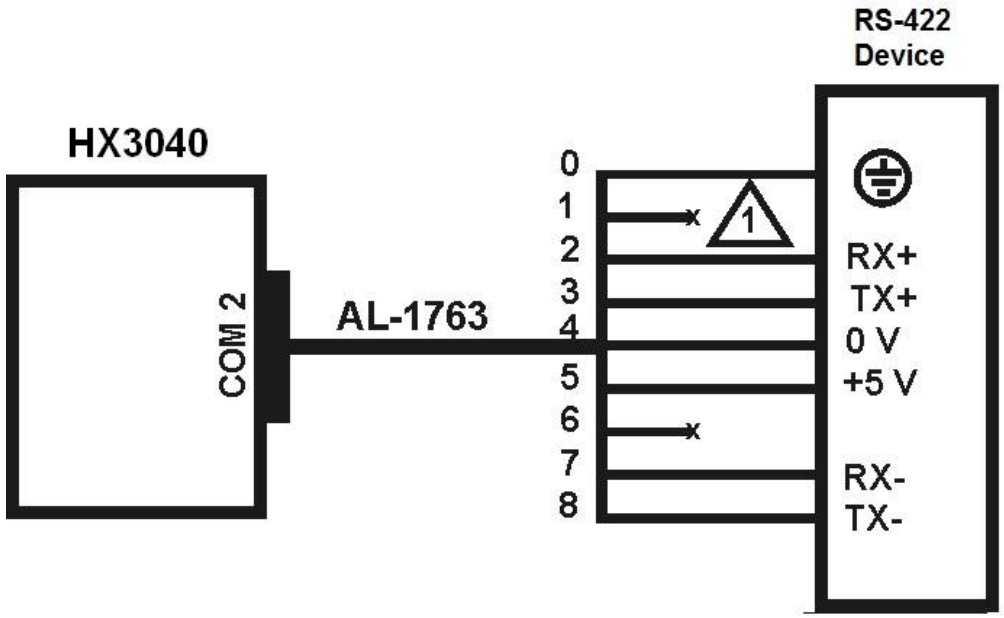


Figura 3-31. Diagrama de Conexão da COM 2 com RS-422 com Terminação Externa

Nota do Diagrama:

Os terminais não conectados devem ser isolados para que não haja contato entre os mesmos.

Exemplo de Rede RS-422

A Figura 3-32 mostra um exemplo de utilização da rede RS-422, utilizando a UCP Xtorm como mestre, dispositivos escravos com interface RS-422, e solução Altus para derivadores e terminadores.

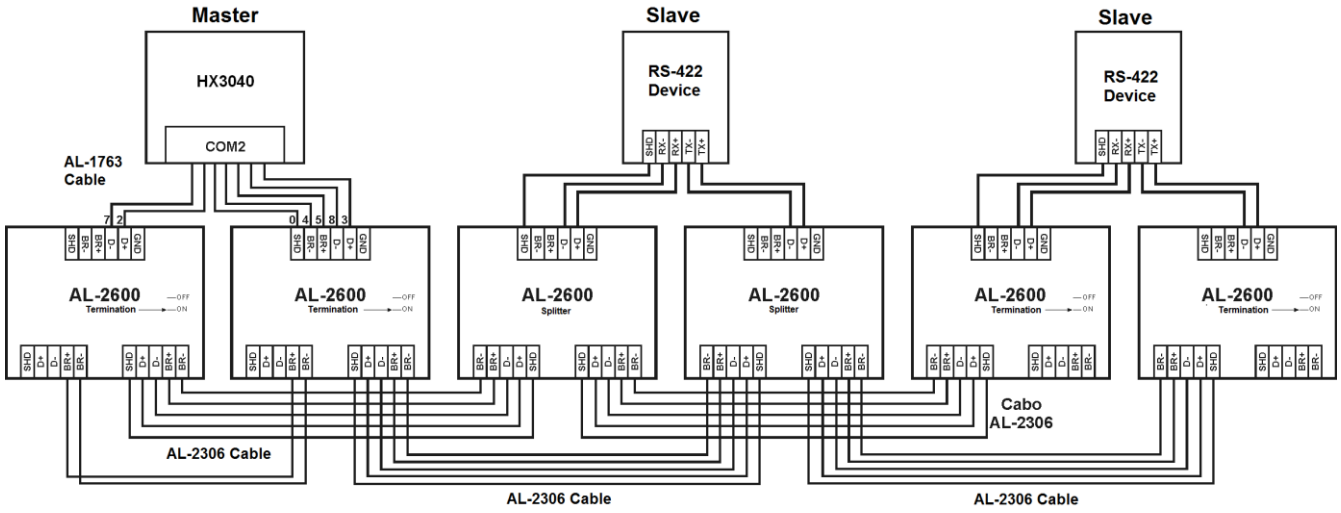


Figura 3-32. Exemplo de Rede RS-422

Nota do Diagrama:

Os módulos AL-2600 que estão nas extremidades da rede fazem a função de terminadores. Neste caso deve-se configura as chaves do AL-2600 em Terminação PROFIBUS.

Nota:

Todos os métodos de terminação são polarizados para comunicação RS-485 e RS-422 – não há terminações passivas.

Conexão com Sinal IRIG-B

Para aplicações onde é necessário o sincronismo de tempo com outros equipamentos, a UCP da Série Hadron Xtorm possui uma porta de entrada e uma porta de saída para o sinal IRIG-B. Através da porta de entrada, a UCP irá receber os dados de sincronismo de tempo e sincronizar com o seu relógio interno. E através da porta de saída, a UCP Xtorm poderá sincronizar outros equipamentos, retransmitindo na porta de saída, o sinal recebido na porta de entrada. Para os casos em que for utilizada a expansão de barramento da Série Xtorm, o sincronismo do relógio interno da UCP se encarregará de sincronizar as estampas de tempo para a geração de eventos dos módulos de E/S presentes na expansão de barramento, conforme a Figura 3-33.

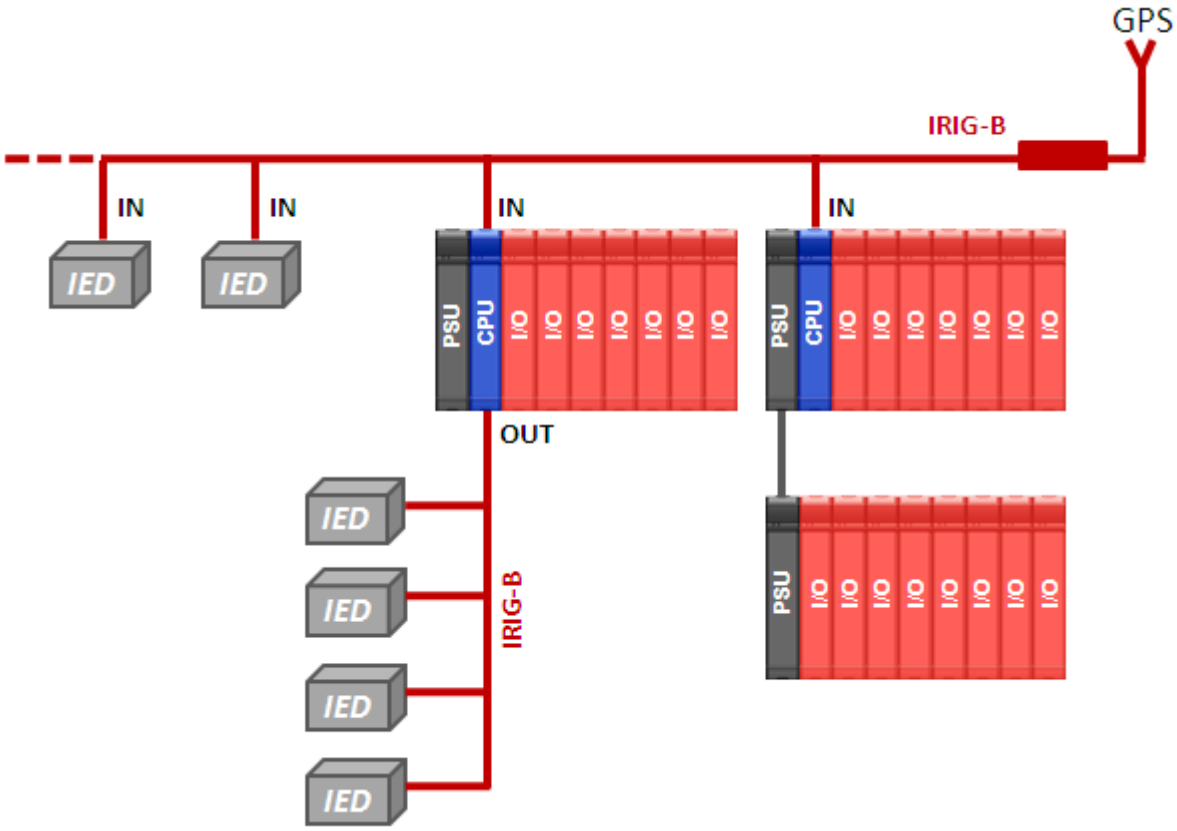


Figura 3-33. Exemplo de Sincronismo para Sistema com Expansões

Para os casos de aplicações onde for utilizada redundância de UCP no mesmo bastidor, o sinal de sincronismo de tempo gerado pelo GPS deve ser conectado na entrada IRIG-B de cada uma das UCPs, independente de qual UCP esteja em modo ativo e qual esteja em modo reserva, garantindo assim o sincronismo de tempo do sistema. Caso seja necessário que a UCP execute a função de repetidora do sinal IRIG-B para outros IEDs, deve-se utilizar a saída de uma UCP somente. Na Figura 3-34 podemos visualizar um exemplo deste tipo de arquitetura.

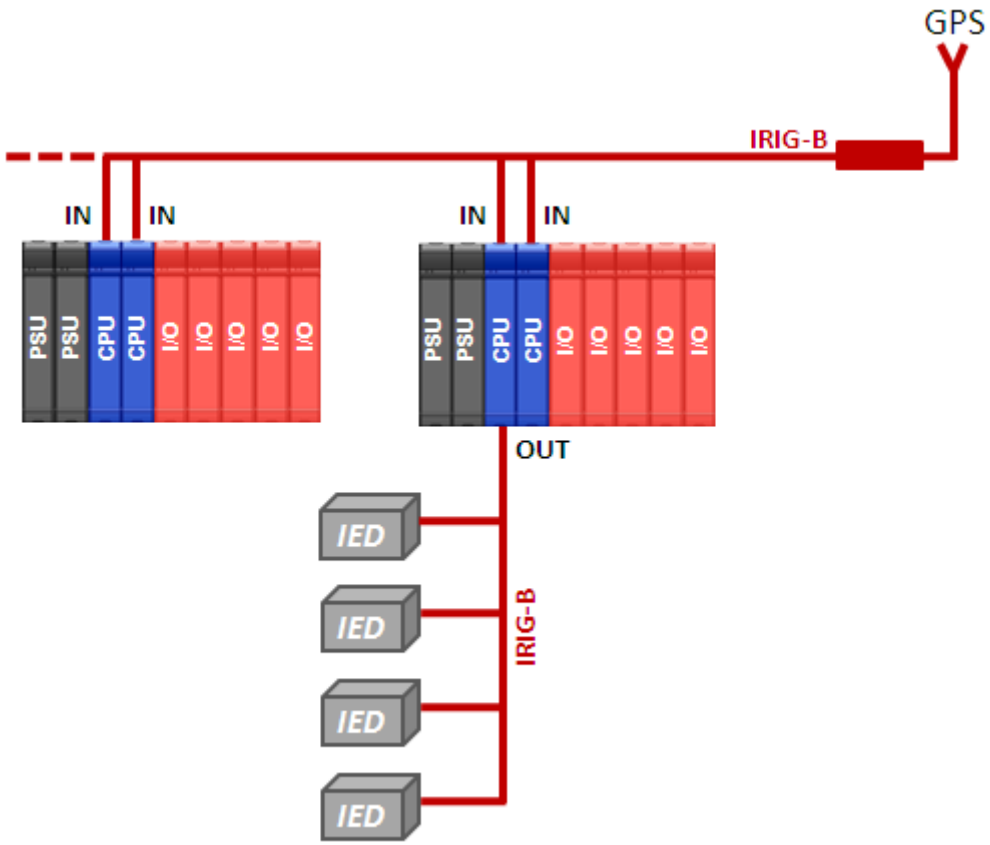


Figura 3-34. Exemplo de Sincronismo para Sistema Redundante

Terminais de Ligação do IRIG-B

Para conexão a um sinal de sincronismo de tempo IRIG-B deve-se conectar os terminais do cabo contendo o sinal IRIG-B nos respectivos bornes da UCP HX3040, conforme a Tabela 3-18.

Nomenclatura	Borne da UCP
IRIG-B +IN	1
IRIG-B -IN	2
IRIG-B +OUT	3
IRIG-B -OUT	4

Tabela 3-18. Conexão dos Sinais IRIG-B na UCP HX3040

ATENÇÃO: Curtos-circuitos prolongados nos conectores do IRIG-B podem danificar a saída do mesmo.

Na Figura 3-35 podemos visualizar a representação do conector IRIG-B da UCP HX3040.

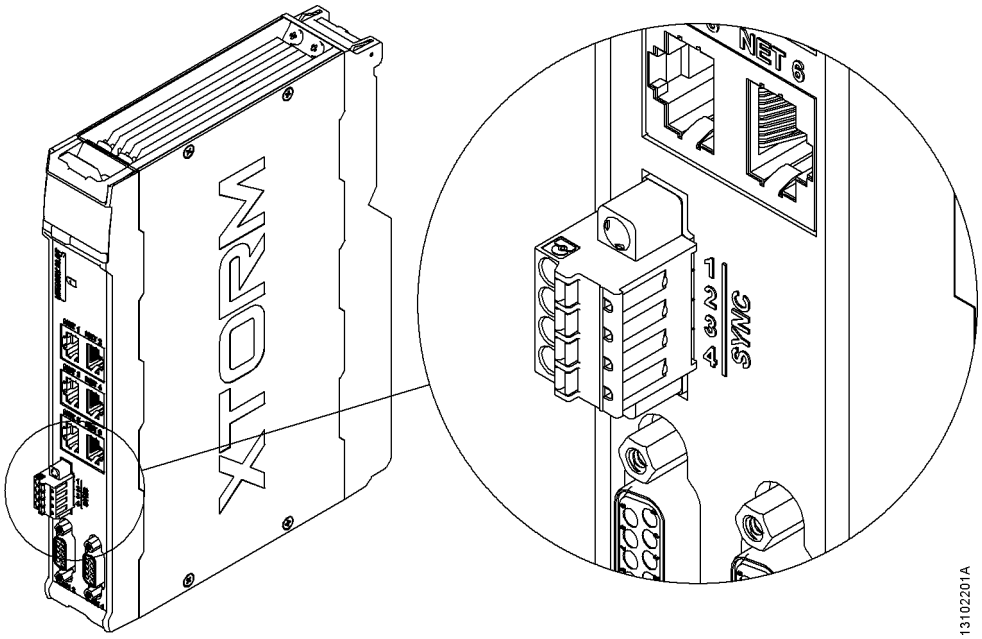


Figura 3-35. Conector de Interligação para o Sinal IRIG-B

Instalação do Cartão de Memória

Nesta seção está descrita a maneira como o cartão de memória deve ser inserido na UCP da Série Hadron Xtorm, sendo que, para maiores informações sobre a utilização do mesmo, consultar o capítulo Configuração – Cartão de Memória.

Inicialmente, deve-se ter atenção quanto à posição correta que o cartão de memória deve ser inserido. Um dos cantos se diferencia dos demais e o mesmo deverá ser utilizado como referência para a inserção correta do cartão. Sendo assim, o cartão de memória deverá ser colocado de acordo com o desenho localizado na parte frontal da UCP ou também como mostra a Figura 3-36.

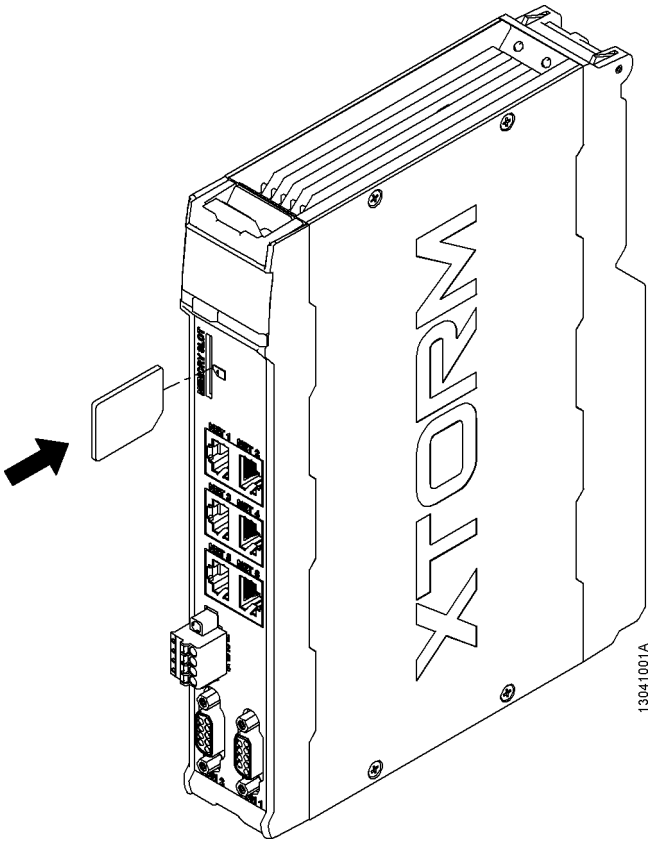


Figura 3-36. Inserção do Cartão de Memória na UCP

Quando o cartão estiver instalado corretamente aparecerá um símbolo no visor gráfico da UCP. Para remover o cartão de memória, basta acessar o Menu "Cartão de Memória" na UCP, utilizando o botão de diagnósticos, e selecionar a opção de remoção do cartão de memória, e aguardar até que o ícone do cartão desapareça da tela de status do visor gráfico.

Para que isso seja possível, deve-se pressionar o cartão contra a UCP até que seja gerado um clique. Então, basta soltá-lo e retirá-lo do compartimento, conforme mostra a Figura 3-37, pois o mesmo estará solto.

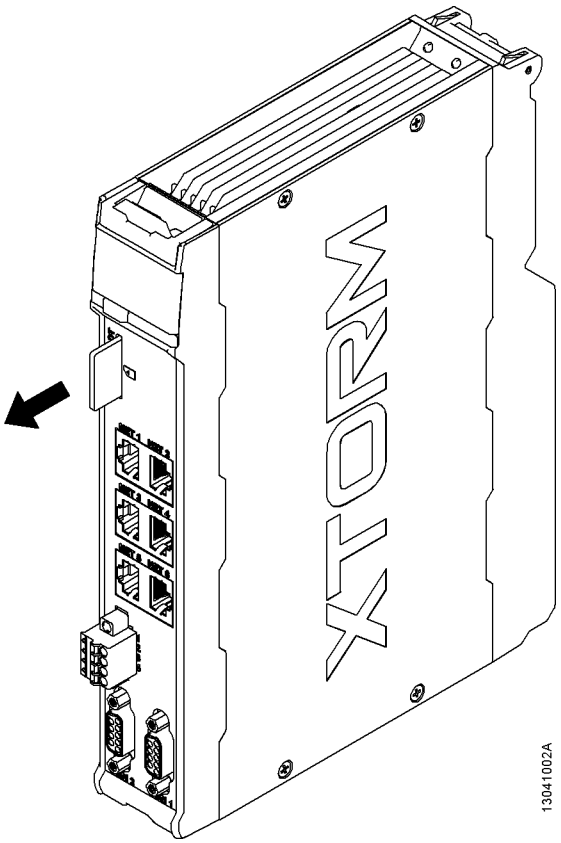


Figura 3-37. Retirada do Cartão de Memória

Instalação do Programador

Para realizar a instalação do software de desenvolvimento MasterTool Xtorm, é necessário ter em mãos o CD-ROM de distribuição ou efetuar o download do arquivo de instalação no site http://www.altus.com.br. Após, feche todos os programas que estejam em execução no seu computador e em seguida dê um duplo clique no arquivo de instalação. O instalador abrirá a seguinte tela de instalação:

Figura 3-38. Assistente de Instalação

A primeira tela indica o início da instalação e apresenta a versão do software MasterTool Xtorm que será instalada em seu computador. Clique em Next para continuar. A tela que se refere ao contrato de licença será mostrada, o qual deve ser lida atentamente. Caso você concorde com os termos da licença, clique em Avançar para continuar.

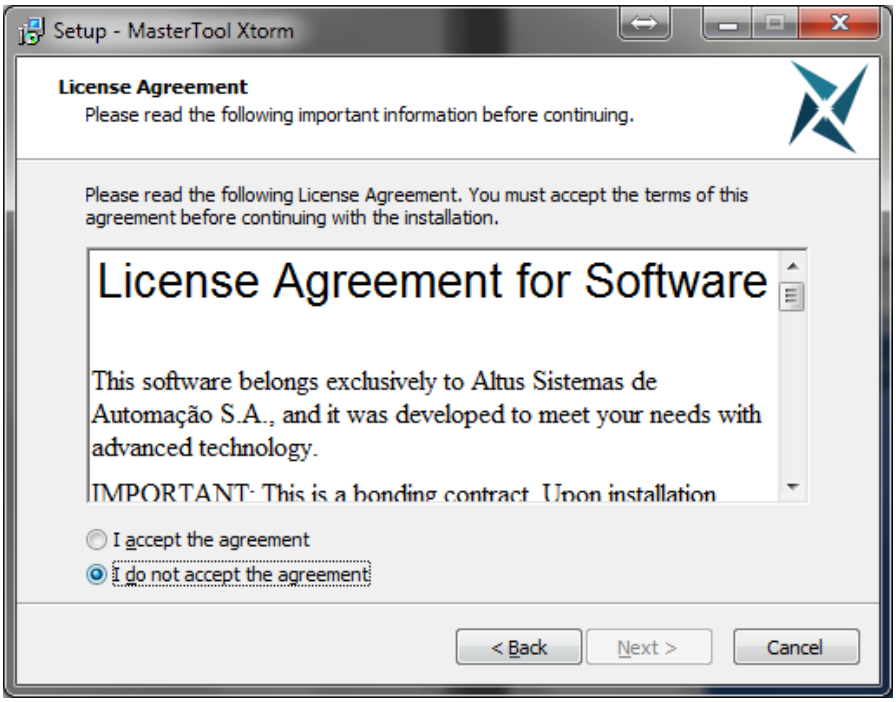


Figura 3-39. Tela de Licença

Na próxima tela é mostrado o programa que será instalado bem como a versão. Clique em Next para continuar.

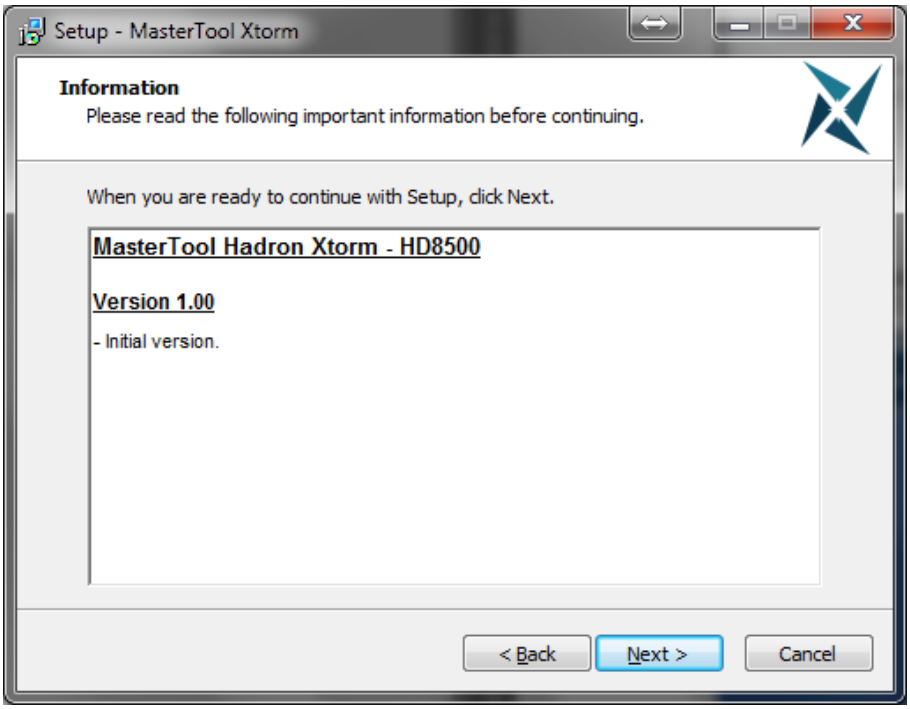


Figura 3-40. Seleção do Caminho de Instalação

Na próxima tela são apresentadas as opções de componentes que serão instalados. Clique em Next para continuar.

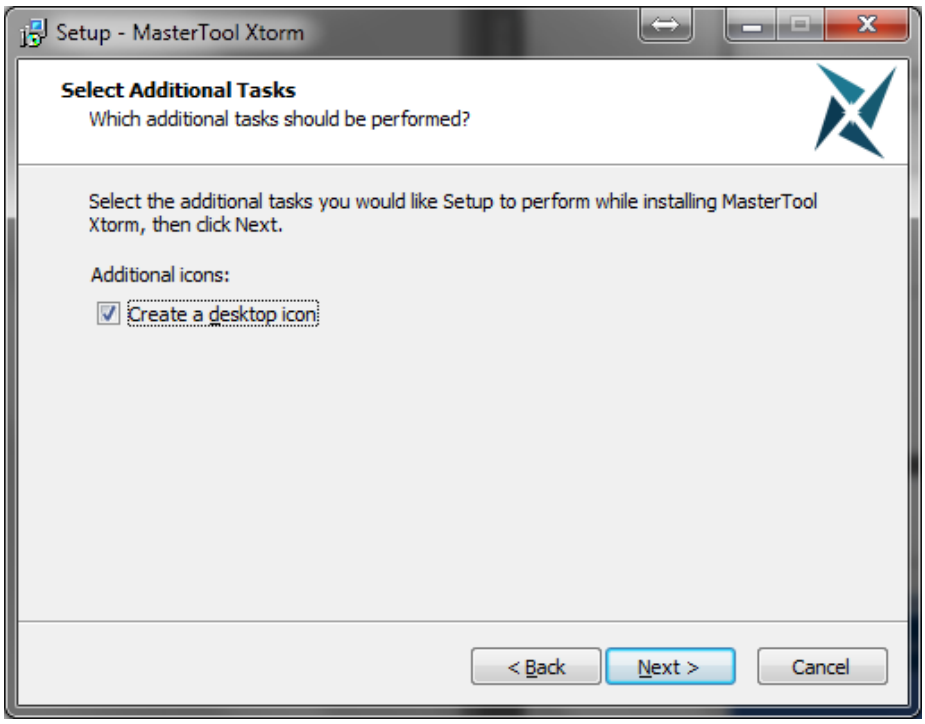


Figura 3-41. Seleção dos Componentes

Surgirá a tela que se refere a instalação ao nome da pasta onde serão mostrados os caminhos e localizações de todos itens a serem instalados. Após, clique em Next para continuar.

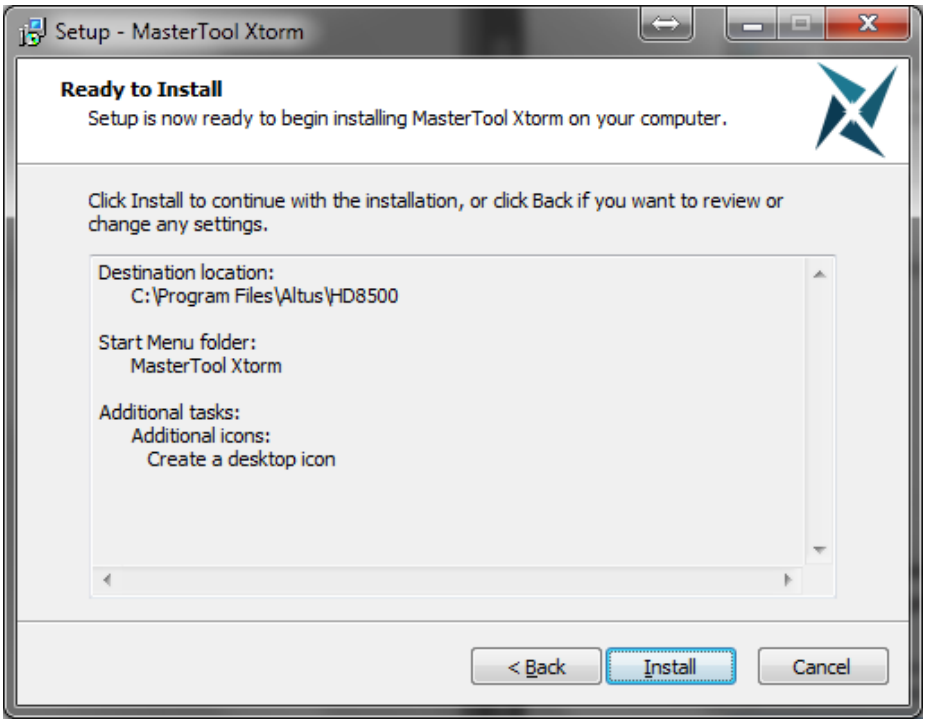


Figura 3-42. Seleção da Pasta

Nessa etapa a instalação do MasterTool Xtorm foi iniciada. Aguarde enquanto os arquivos necessários são instalados em seu computador, isso pode levar alguns minutos dependendo da configuração de seu computador.

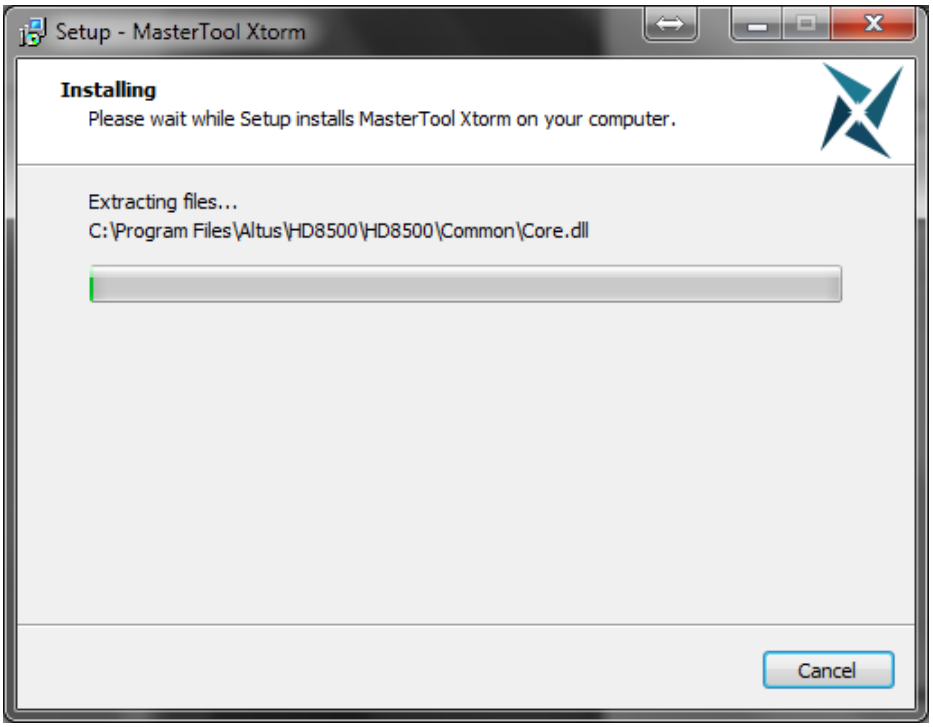


Figura 3-43. Processo de Instalação

Ao fim da instalação será perguntado se o usuário deseja reinicializar o computador para que as modificações da instalação sejam plicadas:

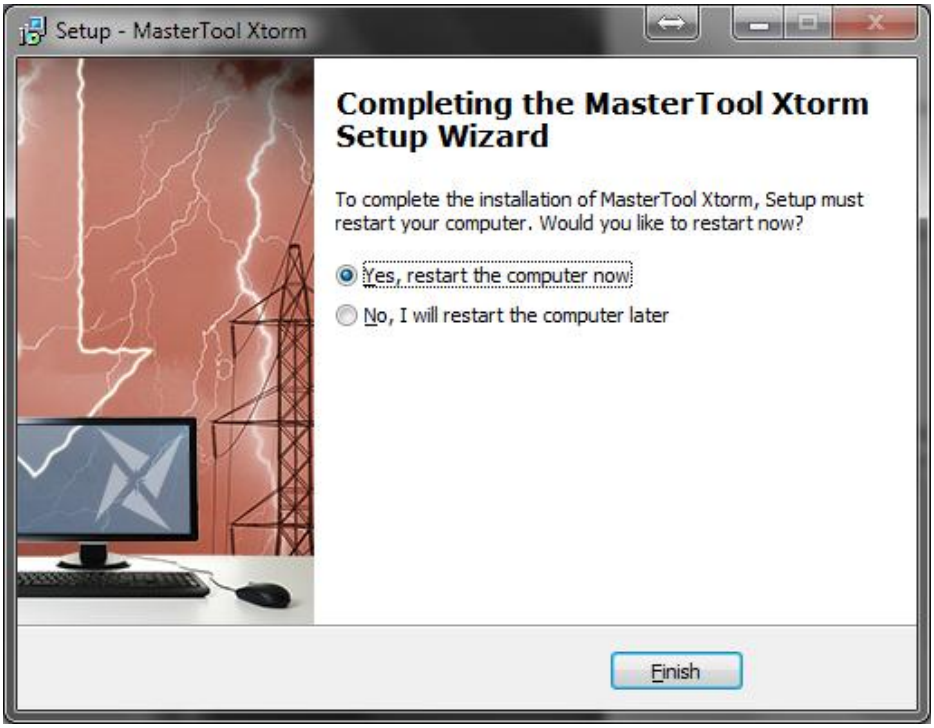


Figura 3-44. Instalação Completa

O MasterTool Xtorm está instalado e pronto para ser utilizado. Para executá-lo, clique no atalho "MasterTool Xtorm" dentro do grupo "Altus" →"MasterTool Xtorm", criado durante a instalação, no menu Iniciar.

Na primeira vez que o software for iniciado, uma tela solicitando as informações de registro será apresentada. Após preencher os campos corretamente, clique em Confirm para utilizar o MasterTool Xtorm.

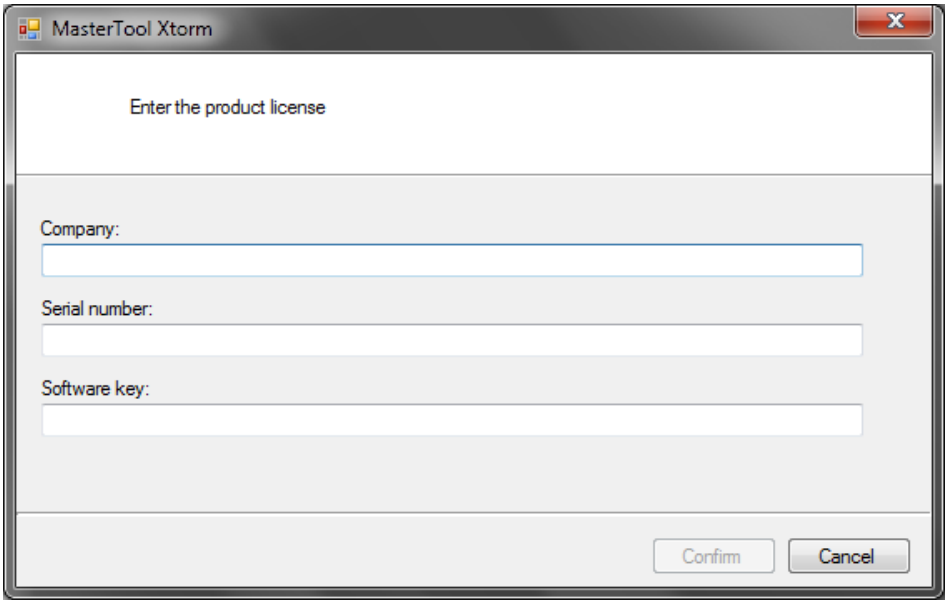


Figura 3-45. Tela de Registro

Para maiores informações sobre a instalação ou utilização do software, consultar o Manual de Programação IEC 61131 - MP399048.

4. Configuração

A UCP da Série Hadron Xtorm é configurada e programada através do software MasterTool Xtorm. A configuração realizada define o comportamento e modos de utilização dos periféricos e características especiais da UCP. A programação representa a aplicação desenvolvida pelo usuário, também chamada de aplicativo.

Configuração da UCP

Os parâmetros relacionados abaixo fazem parte da configuração da UCP inserida na aplicação. Cada item deve ser devidamente revisado para o correto funcionamento do projeto.

Além destes parâmetros, é possível alterar o nome e a descrição de cada módulo inserido na aplicação, para isto, clique com o botão direito sobre o módulo, no item “Propriedades”, na guia “Comum”, altere o nome ou a descrição sendo ambos limitados a 255 caracteres.

As regras e orientações sobre tags, descrições e nome dos módulos de E/S podem ser consultadas na seção “Acessando a descrição do módulo e dos pontos de E/S”.

ATENÇÃO:
É recomendável que o nome de cada módulo possua apenas os caracteres alfanuméricos (maiúsculos ou minúsculos e o texto não iniciar com números). Para a descrição o recomendável é os caracteres alfanuméricos (maiúsculos ou minúsculos), o espaço em branco e o caractér ponto “.”. A utilização de qualquer outro caractér diferente dos acima citados não é recomendada.

ATENÇÃO:
Quando utilizar a função ETD – Electronic Tag on Display para visualizar a descrição do módulo no visor gráfico, a descrição do módulo será truncada nos primeiros 48 caracteres da descrição dada para o módulo.

ATENÇÃO:
Quando utilizar a função ETD – Electronic Tag on Display, Diagnostic Explorer ou Web Server para visualizar o nome do módulo, o nome do módulo será truncado nos primeiros 24 caracteres da descrição dada para o nome do módulo.

Parâmetros Gerais

Os parâmetros gerais (General Parameters), encontrados na tela inicial de configuração da UCP Figura 4-1, são definidos como:

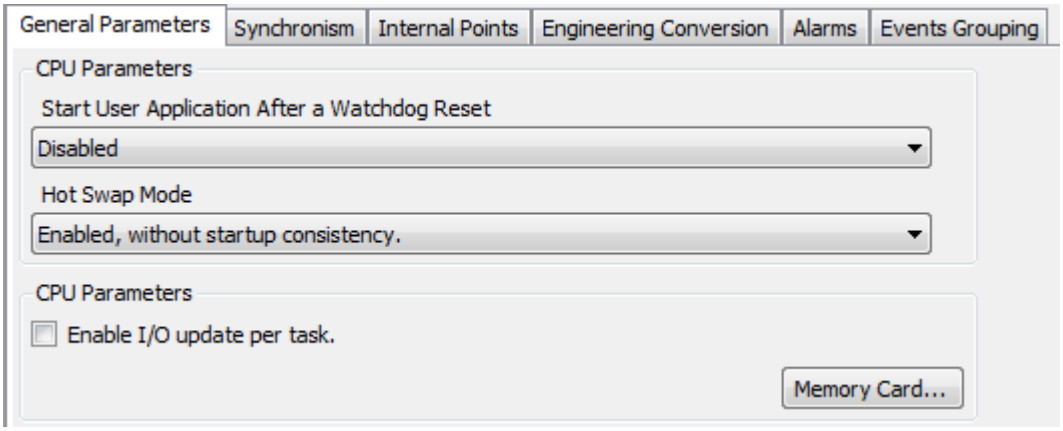


Figura 4-1. Tela de Configuração Parâmetros Gerais da UCP

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Start User Application After a Watchdog Reset	Quando habilitado, inicia a aplicação do usuário após o reset do cão-de-guarda de hardware ou pela reinicialização do Runtime, porém mantêm a indicação através do LED de diagnóstico (botão) e através das variáveis.	Disabled	Enabled Disabled
Hot Swap Mode	Modo troca a quente dos módulos.	Habilitada, sem consistência na partida.	- Disabled for Declared Modules Only: Configuração desabilitada para os módulos que foram declarados na configuração. - Enabled, with Startup Consistency for Declared Modules Only: Configuração habilitada, com consistência na partida apenas para os módulos declarados na configuração. - Enabled,Without Startup Consistency: Configuração habilitada, sem consistência na partida
Enable I/O update per task	Quando habilitado, realiza atualização de entradas e saídas no contexto da tarefa em que estiverem sendo utilizados. Para mais informações, consultar capítulo Escaneamento de E/S.	Disabled	Enabled Disabled

Tabela 4-1. Configurações Gerais da UCP

Troca a Quente

A UCP da Série Hadron Xtorm apresenta a possibilidade de troca dos módulos de E/S do barramento sem a necessidade de desligamento do sistema e sem perda de informações. Esta característica é conhecida como troca a quente.

CUIDADO:
As UCPs da Série Xtorm não garantem a retentividade das variáveis persistentes e retentivas caso a fonte de alimentação, ou a própria UCP, seja removida do bastidor energizado.

Na troca a quente, o comportamento do sistema relacionado se modifica conforme a configuração definida pelo usuário, que apresenta as seguintes opções, conforme descrito na Tabela 4-1:

- Disabled for Declared Modules Only
- Enabled, with Startup Consistency for Declared Modules Only
- Enabled, Without Statup Consistency

Assim, o usuário pode escolher o comportamento que o sistema deverá apresentar em situações anormais de barramento e quando a UCP estiver em Modo Run. A Tabela 4-2 apresenta as possíveis situações anormais de barramento.

Situação	Possíveis causas
Configuração incompatível	- Algum módulo presente no barramento é diferente do modelo que está declarado na configuração.
Módulo ausente	- O módulo foi retirado do barramento. - Algum módulo não está respondendo à UCP por estar com defeito. - Alguma posição do bastidor está com defeito.

Tabela 4-2. Situações Anormais de Barramento

Para maiores informações sobre os diagnósticos correspondentes às situações descritas acima, consultar capítulo Manutenção – Diagnósticos via Variáveis.

Hot Swap Disabled, For Declared Modules Only

Nesta configuração, quando ocorre uma situação anormal de barramento (conforme a Tabela 4-2) a UCP entra em Modo Stop em um tempo de até 2 segundos, quando o LED vermelho começa a piscar 4x (conforme a Tabela 4-3). Para que a UCP volte ao estado normal Run, além de desfazer o que causou a situação anormal, é necessário executar um Reset de qualquer tipo (isso pode ser feito pelo MasterTool Xtorm no menu Comunicação).

ATENÇÃO:
Esta opção não está disponível para projetos com redundância de UCP. Para projetos sem redundância de UCP, a utilização da troca a quente desabilitada também não é indicada quando se utiliza fontes de alimentação redundantes.

Hot Swap Enabled, with Startup Consistency for Declared Modules Only

É considerada "partida" o intervalo entre a energização da UCP (ou comando de reset ou download de aplicação) até a primeira vez em que a mesma entra em modo Run. Esta configuração verifica se ocorreu alguma situação anormal de barramento durante a partida; em caso positivo, a UCP entra em Modo Stop e o LED DL começa a piscar 4x (conforme a Tabela 4-3). Posteriormente, para que a UCP possa ser colocada em modo Run, além de corrigir o que ocasionou a situação anormal, é necessário executar um comando de Reset de qualquer tipo, que pode ser feito pelo MasterTool Xtorm (menu Comunicação).

Após a partida, se algum módulo apresentar alguma das situações citadas na tabela anterior, o sistema continuará trabalhando normalmente e sinalizará o problema via diagnóstico.

ATENÇÃO:
- Nesta configuração, quando ocorrer falta de alimentação (mesmo que temporária), comando Reset a Quente, comando Reset a Frio ou ter sido realizado o Download de uma nova aplicação, e algum módulo estiver em uma situação anormal de barramento; a UCP entrará em Modo Stop e o LED DL começa a piscar 4x (conforme a Tabela 4-3), pois estas são consideradas situações de partida.
- Esta é a opção mais recomendada, pois garante a integridade do sistema na sua inicialização e permite a troca de módulos com o sistema funcionando.

Hot Swap Enabled, without Startup Consistency

Permite que o sistema entre em operação mesmo quando algum módulo estiver em uma situação anormal de barramento (conforme Tabela 4-2). As situações anormais são relatadas via diagnóstico, tanto durante, como após a partida.

ATENÇÃO:
Esta opção é recomendada para a fase de implantação do sistema, pois permite que cargas de novas aplicações e o desligamento da alimentação sejam feitos sem a presença de todos os módulos configurados.

Como realizar a Troca a Quente

CUIDADO:
Antes de proceder à troca a quente, é importante descarregar eventuais potenciais estáticos acumulados no corpo. Para isso, toque (com as mãos nuas) em uma superfície metálica aterrada antes de manipular os módulos. Tal procedimento garante que os níveis de eletricidade estática suportados pelo módulo não serão ultrapassados.

ATENÇÃO:
Recomenda-se o monitoramento dos diagnósticos de troca a quente na aplicação de controle desenvolvida pelo usuário, a fim de garantir que o valor retornado pelo módulo seja validado antes de ser utilizado.

O procedimento para a troca de módulos a quente está descrita no subcapítulo Remoção de Módulos e subcapítulo Inserção dos Módulos.

No caso de módulos de saída, é conveniente que os pontos estejam desligados por ocasião da troca, a fim de reduzir a geração de arcos no conector do módulo. Isso pode ser feito pelo desligamento da fonte de campo ou pelo forçamento dos pontos via ferramentas de software. Se a carga for pequena, não há a necessidade de desligar os pontos.

No caso de troca a quente de fontes de alimentação, é conveniente que a alimentação externa do módulo seja desligada antes de remover o módulo do barramento, a fim de reduzir a geração de arcos no conector do módulo. Para a inserção do módulo fonte de alimentação no barramento, é recomendado que primeiro seja feito o encaixe mecânico do módulo no barramento e em seguida a conexão do conector da alimentação externa ao módulo.

É importante salientar que, nos casos em que a UCP entra em Modo Stop e o LED DL começa a piscar 4x (conforme a Tabela 4-3), devido a alguma situação anormal de barramento (conforme a Tabela 4-2); os módulos de saída colocam seus pontos em modo seguro de falha, ou seja, desligam o ponto (0 Vdc).

No caso dos módulos de entrada digitais, caso o mesmo seja removido do barramento energizado, o estado lógico dos pontos permanecerão no último valor. Caso o(s) conector(es) sejam removidos, o estado lógico dos pontos serão colocados em estado seguro, ou seja, zero.

No caso dos módulos de saída digitais, caso o mesmo seja removido do barramento energizado, o estado lógico dos pontos não serão alterados. Porém, o ponto físico será desenergizado. Caso o(s) conector(es) sejam removidos, o estado lógico dos pontos não será alterado. Porém, o ponto físico será desenergizado.

Para os módulos de entradas analógicas RTD, caso o mesmo seja removido do barramento energizado, o estado lógico dos pontos permanecerão no último valor. Caso o(s) conector(es) sejam removidos, o estado lógico dos pontos irá para o valor de fundo de escala configurado para cada ponto, visto que a impedância de entrada de cada canal sem o conector, tenderá para o infinito.

Para os módulos de entradas analógicas tensão/corrente, caso o mesmo seja removido do barramento energizado, o estado lógico dos pontos permanecerá no último valor. Caso o(s) conector(es) sejam removidos, o estado lógico dos pontos dependerá do tipo de configuração de cada ponto, que vai

depender do tipo de entrada selecionado e dos valores configurados para escala de engenharia. Para a escala de corrente 4-20 mA o estado lógico do ponto dependerá do valor configurado no campo "Valor de Circuito Aberto".

ATENÇÃO:
Proceda sempre à substituição de um módulo por vez, para que a UCP atualize os estados dos módulos.

A Tabela 4-3 relaciona as condições de barramento e o estado de operação e do LED DL da UCP Hadron Xtorm. Para maiores informações sobre os estados do LED de diagnóstico, consultar o capítulo Manutenção – Diagnósticos via LED.

Condição	Habilitada, com consistência na partida apenas para módulos declarados	Habilitada, sem consistência na partida	Desabilitada, apenas para módulos declarados
Módulo não declarado	LED DL: Pisca 2x Azul Aplicação: Run	LED DL: Pisca 2x Azul Aplicação: Run	LED DL: Pisca 2x Azul Aplicação: Run
Módulo ausente	LED DL: Pisca 2x Azul Aplicação: Run	LED DL: Pisca 2x Azul Aplicação: Run	LED DL:Pisca 4x Vermelho Aplicação: Stop
Módulo ausente (condição partida)	LED DL:Pisca 4x Vermelho Aplicação: Stop	LED DL: Pisca 2x Azul Aplicação: Run	LED DL:Pisca 4x Vermelho Aplicação: Stop
Configuração incompatível	LED DL: Pisca 2x Azul Aplicação: Run	LED DL: Pisca 2x Azul Aplicação: Run	LED DL:Pisca 4x Vermelho Aplicação: Stop
Configuração incompatível (condição partida)	LED DL:Pisca 4x Vermelho Aplicação: Stop	LED DL: Pisca 2x Azul Aplicação: Run	LED DL:Pisca 4x Vermelho Aplicação: Stop
Endereço de slot duplicado	LED DL:Pisca 4x Vermelho Aplicação: Stop	LED DL:Pisca 4x Vermelho Aplicação: Stop	LED DL:Pisca 4x Vermelho Aplicação: Stop
Módulo não operacional	LED DL:Pisca 4x Vermelho Aplicação: Stop	LED DL:Pisca 4x Vermelho Aplicação: Stop	LED DL:Pisca 4x Vermelho Aplicação: Stop

Tabela 4-3. Relação entre Condições e Troca a Quente

Cartão de Memória

Os parâmetros encontrados na tela de configuração da cartão de memória Figura 4-52, são definidos como:

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Copiar Projeto da UCP para o Cartão de Memória	Copia o projeto da memória interna da UCP para o cartão de memória	Desabilitado	- Habilitado: Configuração habilitada - Desabilitado: Configuração desabilitada
Senha para Copiar Projeto da UCP para o Cartão de Memória	Senha para copiar o projeto da memória interna da UCP para o cartão de memória	-	Senha de 6 dígitos (0 a 999999)
Copiar Projeto do Cartão de Memória para a UCP	Copia o projeto do cartão de memória para a memória interna da UCP	Desabilitado	- Habilitado: Configuração habilitada - Desabilitado: Configuração desabilitada
Senha para Copiar Projeto do Cartão de Memória para a UCP	Senha para copiar o projeto do cartão de memória para a memória interna da UCP	-	Senha de 6 dígitos (0 a 999999)

Tabela 4-4. Parâmetros de Configuração do Cartão de Memória

Após configurar as possibilidades de cópia do projeto e de ter criado a aplicação de inicialização, deve-se localizar o arquivo "Application.crc" para que as configurações referentes ao cartão de memória tenham efeito. A localização pode ser realizada em Selecionar o "Application.crc" através do botão "Localizar Arquivo...", como pode ser visto na Figura 4-52.

Sincronização de Tempo

Para a sincronização de tempo, a UCP HX3040 usa o protocolo IRIG-B, SNTP (Simple Network Time Protocol) ou o DNP3. Sendo que para cada um deles, o usuário terá a opção de habilitar/desabilitar o protocolo, como também escolher a prioridade de recebimento do sinal de sincronismo dentre cada um dos protocolos.

Para a utilização de protocolos de sincronismo de tempo, o usuário deve configurar os seguintes parâmetros na aba de Synchronism (Sincronismo), acessado através da UCP HX3040, na árvore de dispositivos:

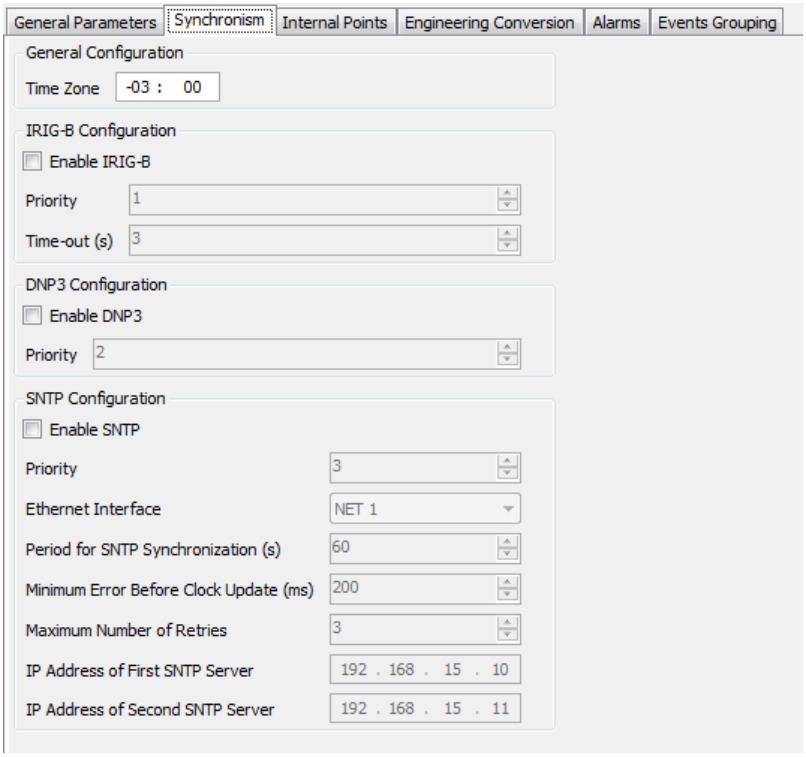


Figura 4-2. Configuração de Sincronismo de Tempo

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Time Zone	Define o tempo que deve ser somado ao horário recebido através das fontes de sincronismos para convertê-lo de UTC para o horário local. A configuração de fuso horário é utilizada em comum entre todas as fontes de sincronismo.	-03:00	-12:59 a +13:59
Enable IRIG-B	Habilita o sincronismo pelo protocolo IRIG-B.	Disabled	Disabled Enabled
Priority	Define a prioridade da fonte de sincronismo com relação às demais. Quando habilitado, o sincronismo pelo protocolo IRIG-B deve exercer prioridade sobre as demais fontes. Devido a isso, o campo está habilitado somente para leitura.	1	1 2 3
Time-Out (s)	Em caso de falha do sinal IRIG-B, define o tempo (em segundos) máximo esperado para a restauração do sinal antes de buscar outra fonte de sincronismo.	3	3 a100
Enable DNP3	Habilita o sincronismo pelo protocolo DNP3.	Disabled	Disabled Enabled
Priority	Define a prioridade da fonte de sincronismo com relação às demais. Quanto menor for o	2	1 2

	valor do campo, mais prioritária é a fonte de sincronismo.		3
Enable SNTP	Habilita o sincronismo pelo protocolo SNTP.	Disabled	Disabled Enabled
Priority	Define a prioridade da fonte de sincronismo com relação às demais. Quanto menor for o valor do campo, mais prioritária é a fonte de sincronismo.	3	1 2 3
Ethernet Interface	Define a porta ethernet pela qual é feito a comunicação com o servidor SNTP usado para a sincronização do relógio.	NET 1	NET 1 NET 2 NET 3 NET 4 NET 5 NET 6
Period for SNTP Synchronization (s)	Período de tempo (em segundos) que deve ser aguardado entre duas solicitações de sincronismo.	60	1 a 255
Minimum Error Before Clock Update (x1 ms)	Antes de atualizar o horário, a UCP compara o horário recebido do servidor SNTP com o seu horário atual. O horário da UCP apenas será atualizado caso a diferença entre o horário recebido e horário atual seja maior que o tempo (em milissegundos) configurado nesse campo.	200	0 a 65535
Maximum Number of Retries	É o número de vezes que a UCP irá tentar estabelecer uma conexão com um dos servidores SNTP. Em caso de erro em todas as tentativas, no próximo período ela irá tentar se conectar com o outro servidor.	3	0 a 100
IP Address of First SNTP Server	Endereço IP do primeiro servidor SNTP a ser utilizado.	192.168.15.10	1.0.0.1 a 223.255.255.254
IP Address of Second SNTP Server	Endereço IP do segundo servidor SNTP a ser utilizado.	192.168.15.11	1.0.0.1 a 223.255.255.254

Tabela 4-5. Configurações de Sincronismo de Tempo

Para a opção de sincronismo de tempo via IRIG-B, é empregada uma tarefa oculta que atualiza o RTC da UTR com o valor de sincronismo de tempo recebido via IRIG-B a cada 10 segundos. Sendo assim, o usuário deve levar em consideração o tempo da tarefa do componente, sendo que o RTC pode ser atualizado até 10 segundos após o recebimento do novo valor para o sincronismo de tempo.

Caso a UTR seja desligada antes que ocorra a atualização do RTC com o novo valor de sincronismo de tempo recebido, esse novo valor será perdido, visto que após a energização do sistema, a UTR irá utilizar o último valor gravado no RTC para a hora do sistema.

No caso do sincronismo via DNP3, o usuário precisa habilitar o sincronismo de tempo através da opção “Enable DNP3” na tela de sincronismo.Também é necessário que exista um driver do protocolo DNP3 Server instanciado, e um dispositivo configurado para enviar os dados de sincronismo. Para configurar o dispositivo com essa opção, verificar os parâmetros “Enable Synchronism” e “Synchronism Time-Out” disponíveis no capítulo Configuração Configuração da Camada de Aplicação Servidor DNP3. O sincronismo de tempo via DNP3 depende da habilitação das duas checkboxes acima descritas, uma vez que, dentre as opções de sincronismo disponíveis na RTU, é necessário habilitar o sincronismo via DNP3 quando necessário, e neste caso, dentre todos os dispositivos configurados para se conectar ao Servidor DNP3, é necessário configurar qual deles estará habilitado a enviar os dados de sincronismo de tempo.

Para o sincronismo via SNTP a UCP irá se comportar como um cliente SNTP, ou seja, enviará requisições de sincronização de tempo para um servidor SNTP/NTP, que pode estar na rede local ou na internet. A precisão do sincronismo de tempo por SNTP depende das características da rede Ethernet onde está sendo aplicado. Caso o SNTP Server esteja na mesma rede local em que está a UTR, em uma situação ideal é possível obter uma precisão de 1ms. Contudo, quando o SNTP Server estiver em uma rede remota (internet, etc.) este número pode chegar a dezenas de milissegundos. Ou

seja, quando uma sincronização for efetuada, o horário atualizado no cliente pode estar alguns milissegundos adiantado ou atrasado em relação ao servidor.

A UCP HX3040 envia as requisições de sincronização cíclicas, de acordo com o tempo configurado no campo Período de Sincronização do SNTP. Na primeira tentativa de sincronização, logo após a inicialização do serviço, a requisição é para o primeiro servidor, configurado em Endereço de IP do 1º Servidor. Caso este não responda, as requisições são direcionadas para o segundo servidor configurado em Endereço de IP do 2º Servidor, fornecendo uma redundância de servidores SNTP. Caso o segundo servidor também não responda, o mesmo processo de tentativa de sincronização é executado novamente, mas apenas após o Período de Sincronização ter passado. Ou seja, a cada período de sincronização, a UCP tenta se conectar uma vez em cada servidor, ela tenta o segundo caso o primeiro não responda. O tempo de espera por uma resposta do servidor SNTP é definido por padrão em 5 segundos e não pode ser modificado.

Caso, após uma requisição de sincronização, a diferença entre o horário atual da UCP HX3040 e o recebido pelo servidor for maior que o valor configurado no parâmetro Erro Mínimo Antes da Atualização do Relógio, o horário da UCP HX3040 é atualizado.

Como o SNTP usa o horário no formato UTC (Universal Time Coordinated), o parâmetro de Fuso Horário deve ser configurado corretamente, para que o horário lido pelo SNTP seja convertido corretamente para a hora local.

O Processo de execução do cliente SNTP pode ser exemplificado com os seguintes passos:

- Tentativa de sincronização através do primeiro servidor. Caso a sincronização ocorra com sucesso, a UCP aguarda o tempo para a nova sincronização (Período de Sincronização) e tentará sincronizar-se novamente com o este servidor, utilizando, então, este como servidor primário. Em caso de falha (o servidor não responde em menos de 5 s) o passo 2 é executado
- Tentativa de sincronização através do segundo servidor. Caso a sincronização ocorra com sucesso, a UCP aguarda o tempo para a nova sincronização (Período de Sincronização) e tentará sincronizar-se novamente com este servidor, utilizando, então, este como servidor primário. Em caso de falha (o servidor não responde em menos de 5 s) é aguardado o tempo referente ao Período de Sincronização e executado novamente o passo 1

Como o tempo de espera pela resposta do servidor SNTP é de 5 s, deve-se prestar atenção ao configurar valores menores do que 10 s para o Período de Sincronização. Caso o servidor primário não responda, o tempo para a sincronização irá ser de, no mínimo, 5 s (aguardo da resposta do servidor primário e tentativa de sincronização com o servidor secundário). Caso nem o servidor primário nem o secundário respondam, o tempo para a sincronização irá ser de, no mínimo, 10 s (aguardo da resposta dos dois servidores e nova tentativa de conexão com o 1º servidor).

Dependendo da subrede do servidor SNTP, o cliente irá utilizar a interface Ethernet que esteja na subrede corresponde para fazer as requisições de sincronismo. Caso o servidor esteja fora das duas subredes das interfaces NET 1 e NET 2, a requisição poderá ser feita por qualquer uma das duas interfaces que possa achar uma rota para o servidor.

Para a utilização do cliente SNTP, o usuário deve configurar os seguintes parâmetros na aba Configuração de SNTP, acessado através da UCP HX3040, na árvore de dispositivos:

Notas:

Servidor SNTP: É possível definir um endereço preferencial e outro secundário para acessar um servidor SNTP e, assim, obter sincronismo de tempo. Se ambos os campos estiverem vazios, o serviço SNTP permanecerá desabilitado.

Visualização dos Diagnósticos: Caso nenhuma sincronização de tempo tenha ocorrido entre o momento do envio do projeto e o momento atual, os diagnósticos do SNTP, presentes nos diagnósticos detalhados da UCP estarão com o valor “0” e somente mudarão assim que ocorrer uma atualização de horário, ou seja quando o parâmetro de tempo mínimo para atualização do relógio exceder a diferença de desacerto do relógio entre a UCP e o Servidor.

Minimum Error Before Clock Update: Caso este parâmetro seja igual à zero, o relógio será atualizado em todas as requisições de sincronismo.

ATENÇÃO:
Alterações do fuso-horário em tempo de execução quando o método de sincronismo de tempo é IRIG-B não são recomendadas. Estas alterações podem causar descontinuidades em estampas de tempo, que por sua vez causariam um comportamento não desejado na geração de certos eventos.

ATENÇÃO:
O Serviço SNTP depende da aplicação do usuário apenas para a sua configuração. Portanto, este serviço vai ser executado mesmo quando a UCP estiver nos modos STOP ou BREAKPOINT, desde que exista uma aplicação na UCP com o cliente SNTP habilitado e corretamente configurado.

Em arquiteturas com redundância de UCP, o driver DNP3 Server é desabilitado na UCP não ativa. Desta forma, não é recomendada a utilização deste método de sincronismo em sistemas redundantes, pois a UCP não ativa pode levar vários segundos após um switchover até que o seu relógio seja sincronizado. Em sistemas redundantes, recomenda-se a utilização de SNTP e/ou IRIG-B.

CUIDADO:
É vital a configuração de, no mínimo, um servidor SNTP. O recomendável é configurar dois servidores SNTP (primário e secundário). O sincronismo SNTP é necessário para gerar eventos com estampa de tempo coerente entre os UCPA e UCPB e com a hora mundial. Outra utilidade é evitar descontinuidades durante um switchover em aplicações que referenciam data e hora, tendo em vista que não existe sincronismo de data e hora entre as UCPs através dos canais de sincronismo das NETs da UCPA e UCPB.

CUIDADO:
Para aplicações onde a sincronização de tempo necessita de uma precisão na ordem de poucos milissegundos, sem interferência da tarefa principal, recomenda-se o uso do protocolo IRIG-B.

Horário de Verão

A configuração do horário de verão deve ser feita indiretamente, através da função SetTimeZone, que altera o fuso horário aplicado ao RTC. No início do horário de verão, deve-se usar a função para aumentar em uma hora o fuso horário. Ao final do horário de verão, ela é usada novamente para diminui-lo em uma hora.

Para maiores informações, consultar a subseção Relógio RTC - Blocos Funcionais e Funções para Leitura e Escrita do RTC deste manual.

Pontos Internos

Um ponto de comunicação é armazenado na memória da UTR sob a forma de duas variáveis distintas. Uma delas representa o valor do ponto (tipo BOOL, BYTE, WORD, etc...), enquanto a outra representa a sua qualidade (tipo QUALITY). Pontos Internos são aqueles cujo valor e qualidade são calculados internamente pela aplicação do usuário, isto é, não possuem uma origem externa como ocorre para pontos associados à IEDs (drivers de comunicação do tipo Mestre/Cliente) ou a módulos de E/S.

A função desta aba de configuração de Pontos Internos é relacionar a variável que representa o valor de um ponto com a variável que representa a sua qualidade. Deve ser utilizada para relacionar variáveis de valor e qualidade criadas internamente no programa da UTR (como em uma GVL por exemplo), as quais tipicamente serão mapeadas posteriormente para um driver de comunicação do tipo Servidor para comunicação com o centro de controle.

Desta forma, o objetivo desta aba não é criar ou declarar pontos internos. Para realizar isto, basta declarar as variáveis de valor e/ou qualidade em uma GVL e mapeá-la no driver de comunicação.

A configuração dos pontos internos, visualizadas na Figura 4-3, segue os parâmetros descritos na Tabela 4-6. É possível a configuração de até 5120 entradas na tabela de Pontos Internos.

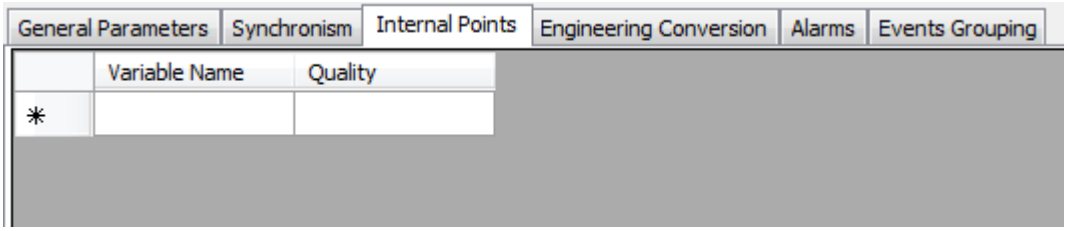


Figura 4-3. Tela de Configuração dos Pontos Internos

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Variable Name	Variável simbólica que armazena o valor do ponto interno.	-	Aceita variáveis do tipo BOOL, WORD, DWORD, LWORD, INT, DINT, LINT, UINT, UDINT, ULINT, REAL, LREAL ou DBP. A variável pode ser simples, array ou elemento de array e pode estar em estruturas.
Quality	Variável simbólica que armazena a qualidade do ponto interno representado em "Nome da Variável".	-	Variáveis do tipo QUALITY (LibRtuStandard), que podem ser simples, array ou elemento de array e podem estar em estruturas.

Tabela 4-6 Configuração dos Pontos Internos

A figura abaixo mostra um exemplo de configuração de um ponto interno:

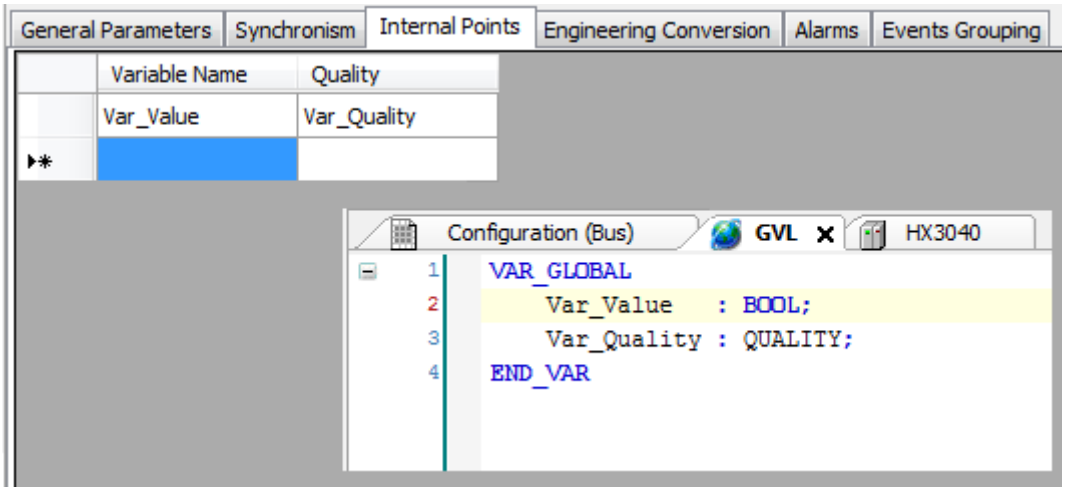


Figura 4-4. Exemplo de Configuração dos Pontos Internos

Conversões de Qualidade

A qualidade de um ponto interno é uma informação que indica o nível de confiança que se pode ter no valor que está armazenado naquele ponto. A qualidade pode informar, por exemplo, que o valor armazenado está fora de escala, ou ainda que seja válido, mas pouco confiável.

As normas IEC 61850 e DNP3 possuem os seus próprios formatos para representação da informação de qualidade de um ponto. A Série Xtorm, por sua vez, possui um formato de qualidade próprio (mas muito similar ao IEC 61850) chamado de Internal Quality. Este formato é definido pelo tipo QUALITY (biblioteca LibRtuStandard) e é utilizado internamente para armazenamento da qualidade, permitindo que sejam realizadas conversões entre protocolos sem que haja perda de informação.

Quando se realiza o mapeamento de um mesmo ponto de comunicação entre dois drivers, a conversão de qualidade é realizada automaticamente em dois estágios. Por exemplo: caso um ponto de comunicação seja mapeado de um driver DNP3 Client para um driver IEC 61850 Server, primeiro a qualidade será convertida do formato DNP3 para o formato interno (e armazenada internamente na UCP), e depois será convertida do formato interno para o formato IEC 61850.

As tabelas a seguir definem as conversões dos formatos proprietários dos protocolos para o formato interno. Caso seja necessário consultar a conversão entre os protocolos, é necessário realizar a análise em dois estágios consultando cada uma das tabelas para o formato interno e depois realizando a correlação entre elas.

Internal Quality

Está é a estrutura QUALITY, a Tabela 4-7 mostra detalhadamente cada uma das suas componentes.

Bit	Nome	Tipo	Descrição
0	FLAG_RESTART	BOOL	A flag de RESTART indica que o dado não foi atualizado pelo campo desde o Reset do dispositivo.
1	FLAG_COMM_FAIL	BOOL	Indica que existe uma falha de comunicação no caminho entre o dispositivo onde o dado é originado e o dispositivo de relatórios.
2	FLAG_REMOTE_SUBSTITUTED	BOOL	Se TRUE, o valor dos dados é substituído em um dispositivo de comunicação remota.
3	FLAG_LOCAL_SUBSTITUTED	BOOL	Se TRUE, o valor do dado é substituído pelo dispositivo que gerou essa flag. Este comportamento pode ter ocorrido devido a um funcionamento em diagnóstico ou temporário devido à intervenção humana.
4	FLAG_FILTER	BOOL	Flag utilizada para sinalizar e prevenir a sobrecarga dos canais de comunicação de eventos, como oscilações (mudanças rápidas) nas entradas binárias.
5	FLAG_OVERFLOW	BOOL	Este identificador deve indicar um problema de qualidade de que o valor do atributo que a qualidade for associada está além da capacidade de representação.
6	FLAG_REFERENCE_ERROR	BOOL	Este identificador deve indicar que o valor pode não ser um valor correto devido à referência estar fora de calibração.
7	FLAG_INCONSISTENT	BOOL	Este identificador deve indicar que uma função de avaliação detectou uma inconsistência.
8	FLAG_OUT_OF_RANGE	BOOL	Este identificador deve indicar um erro de qualidade que o atributo ao qual a qualidade foi associada está além da capacidade de valores pré-definidos.
9	FLAG_INACCURATE	BOOL	Este identificador deve indicar que o valor não atende a precisão declarada da fonte.
10	FLAG_OLD_DATA	BOOL	Um valor parece estar desatualizado, caso uma atualização não ocorra em um período de tempo especifico.
11	FLAG_FAILURE	BOOL	Este identificador deve indicar que uma função de supervisão detectou uma falha interna ou externa.
12	FLAG_OPERATOR_BLOCKED	BOOL	Atualização foi bloqueada pelo operador.
13	FLAG_TEST	BOOL	Test deve ser um identificador adicional que pode ser usado para classificar um valor sendo um valor teste e que não será usado para fins operacionais.
14-15	RESERVED	Reservado	
16-17	VALIDITY	QUALITY_VALIDITY	0 – Good (Valor confiável, significa que não há condição anormal) 1 – Invalid (Valor não corresponde ao valor no IED) 2 – Reserved (Reservado) 3 – Questionable (Valor presente pode não ser o mesmo do IED)

Tabela 4-7. Estrutura QUALITY

Conversão DNP3

A Tabela 4-8 ilustra a conversão de qualidade DNP3 para pontos internos, e a Tabela 4-9 relaciona a conversão de qualidade de pontos internos para DNP3 da série Xtorm disponíveis para HD8500.

DNP3 -> Interno		
DNP3 Quality	Qualidade Interna	
	Flags	VALIDITY
ONLINE	-	IF JUST ONLINE, VALIDITY_GOOD
NOT ONLINE	FLAG_OLD_DATA	VALIDITY_QUESTIONABLE
RESTART	FLAG_RESTART	VALIDITY_QUESTIONABLE
COMM_LOST	FLAG_COMM_FAIL AND FLAG_OLD_DATA	VALIDITY_QUESTIONABLE
REMOTE_FORCED	FLAG_REMOTE_SUBSTITUTED AND FLAG_OPERATOR_BLOCKED	IF JUST REMOTE_FORCED AND ONLINE, VALIDITY_GOOD
LOCAL_FORCED	FLAG_LOCAL_SUBSTITUTED AND FLAG_OPERATOR_BLOCKED	IF JUST LOCAL_FORCED AND ONLINE, VALIDITY_GOOD
CHATTER_FILTER	FLAG_FILTER	VALIDITY_QUESTIONABLE
OVER_RANGE	FLAG_OVERFLOW AND FLAG_OUT_OF_RANGE	VALIDITY_INVALID
DISCONTINUITY	FLAG_REFERENCE_ERROR	VALIDITY_QUESTIONABLE
REFERENCE_ERR	FLAG_INACCURATE AND FLAG_REFERENCE_ERROR	VALIDITY_QUESTIONABLE

Tabela 4-8. Conversão DNP3 para Interno

Interno-> DNP3		
Qualidade Interna		Qualidade DNP3
Flags	VALIDITY	
FLAG_RESTART	ANY	RESTART
FLAG_COMM_FAIL	ANY	COMM_LOST
FLAG_REMOTE_SUBSTITUTED	ANY	REMOTE_FORCED
FLAG_LOCAL_SUBSTITUTED	ANY	LOCAL_FORCED
FLAG_FILTER	ANY	CHATTER_FILTER
FLAG_OVERFLOW	ANY	OVER_RANGE
FLAG_REFERENCE_ERROR	ANY	DISCONTINUITY (para contadores) REFERENCE_ERR (para analógicos)
FLAG_INCONSISTENT	ANY	not ONLINE
FLAG_OUTOFRANGE	ANY	OVER_RANGE
FLAG_INACCURATE	ANY	REFERENCE_ERR
FLAG_OLDDATA	ANY	not ONLINE
FLAG_FAILURE	ANY	not ONLINE
FLAG_OPERATOR_BLOCKED AND NOT LOCAL_SUBSTITUTED OR AND NOT REMOTE_SUBSTITUTED	ANY	LOCAL_FORCED
FLAG_TEST	ANY	not ONLINE
NOT FLAG_OLD_DATA AND NOT FLAG_FAILURE AND NOT FLAG_INCONSISTENT AND NOT FLAG_TEST	VALIDITY_GOOD	ONLINE

Tabela 4-9. Conversão Interno para DNP3

Conversão IEC 61850

A Tabela 4-10 relaciona o tipo de qualidade dos pontos internos com a qualidade IEC 61850 na série Xtorm disponíveis para HD8500.

Interno-> IEC 61850		
Flags de Qualidade Interna	**VALIDITY IEC 61850**	**Flags IEC 61850**
FLAG_RESTART	VALIDITY_QUESTIONABLE	OLDDATA
FLAG_COMM_FAIL	VALIDITY_QUESTIONABLE	OLDDATA
FLAG_REMOTE_SUBSTITUTED	GOOD	Source = SUBSTITUTED
FLAG_LOCAL_SUBSTITUTED	GOOD	Source = SUBSTITUTED
FLAG_FILTER	VALIDITY_QUESTIONABLE OR VALIDITY_INVALID	OSCILLATORY
FLAG_OVERFLOW	VALIDITY_INVALID	OVERFLOW
FLAG_REFERENCE_ERROR	VALIDITY_QUESTIONABLE OR VALIDITY_INVALID	BADREFERENCE
FLAG_INCONSISTENT	VALIDITY_QUESTIONABLE	INCONSISTENT
FLAG_OUT_OF_RANGE	VALIDITY_QUESTIONABLE OR VALIDITY_INVALID	OUTOFRANGE
FLAG_INACCURATE	VALIDITY_QUESTIONABLE	INNACURATE
FLAG_OLD_DATA	VALIDITY_QUESTIONABLE	OLDDATA
FLAG_FAILURE	VALIDITY_INVALID	FAILURE
FLAG_OPERATOR_BLOCKED	GOOD	OPERATORBLOCKED
FLAG_TEST	GOOD	TEST

Tabela 4-10. Conversão Interno para IEC 61850

O processo de conversão do formato de qualidade interno para IEC 61850, segue as recomendações definidas pela norma IEC 61850. Segundo a norma o campo validity é definido em função das flags de qualidade. Algumas flags permitem que a validity assuma tanto o valor VALIDITY_QUESTIONABLE quanto VALIDITY_INVALID, nestes casos a validade assumida é a definida no ponto a ser convertido.

Qualidade Interna MODBUS

Como a norma MODBUS não especifica tipos de qualidade para cada ponto, mas para auxilio no uso dos diagnósticos de comunicação de cada ponto, a Série Hadron Xtorm permite através de uma estrutura interna própria o mapeamento de variáveis de qualidade para cada ponto MODBUS. A Tabela 4-11 descreve os tipos de qualidade que cada ponto MODBUS pode assumir.

Qualidade Resultante	VALIDITY Resultante	Descrição
FLAG_RESTART	VALIDITY_INVALID	Valor inicial, nenhuma comunicação começou ainda
No error flag (0)	VALIDITY_GOOD	Comunicação OK. O ponto está atualizado
FLAG_COMM_FAIL AND FLAG_RESTART	VALIDITY_INVALID	Erro de comunicação. O ponto nunca foi atualizado
FLAG_COMM_FAIL AND FLAG_OLD_DATA	VALIDITY_QUESTIONABLE	Um erro ocorreu mas o ponto estava atualizado e possui um valor antigo.
FLAG_FAILURE AND FLAG_RESTART	VALIDITY_INVALID	Recebeu uma resposta de exceção, e o ponto ainda está com o valor inicial.
FLAG_FAILURE AND FLAG_OLD_DATA	VALIDITY_QUESTIONABLE	Recebeu uma resposta de exceção, mas o ponto possui um valor antigo válido
FLAG_RESTART	VALIDITY_INVALID	Dispositivo está parado. O ponto nunca foi atualizado.
FLAG_RESTART AND FLAG_OLD_DATA	VALIDITY_QUESTIONABLE	Dispositivo está parado. O ponto contém um valor antigo

Tabela 4-11. Qualidade MODBUS

Qualidade dos Módulos de E/S

Para auxiliar no uso dos diagnósticos de cada ponto de E/S, a Série Hadron Xtorm cria automaticamente uma estrutura de qualidade para cada módulo utilizado no projeto da UTR, através de uma estrutura interna própria acessível via structure QUALITY, disponível na GVL IOQualities criada automaticamente pelo template do HD8500. A Tabela 4-12 descreve os tipos de qualidade que cada ponto de entrada e saída.

Para mais informações consulte o capitulo GVL IOQualities.

Diagnóstico	Qualidade Resultante	VALIDITY Resultante	Descrição
Don't care	FLAG_RESTART	VALIDITY_INVALID	A qualidade tem este valor antes de ser lida ou escrita pela primeira vez.
None	No error flag (0)	VALIDITY_GOOD	Comunicação OK. O ponto está atualizado.
None	FLAG_OLD_DATA AND FLAG_FAILURE	VALIDITY_QUESTIONABLE	Módulo não operacional, entretanto dados foram lidos/escritos pelo menos uma vez.
bOverRange OR bUnderRange	FLAG_OUT_OF_RANGE	VALIDITY_INVALID	Over range ou under range em módulo de entrada.
bOpenLoop	FLAG_FAILURE	VALIDITY_INVALID	Laço aberto em módulo de entrada.
bInputNotEnable	FLAG_OPERATOR_BLOCKED	VALIDITY_INVALID	Entrada não está habilitada.
bNoExternalSupply	FLAG_FAILURE	VALIDITY_INVALID	Alimentação externa está abaixo do limite mínimo de funcionamento.
bShortCircuit	FLAG_FAILURE	VALIDITY_INVALID	Curto-circuito na saída
bCalibrationError	FLAG_INACCURATE	VALIDITY_INVALID	Erro de Calibração.
bColdJunctionSensorError	FLAG_INACCURATE	VALIDITY_INVALID	Erro no Sensor de Junta Fria.

Tabela 4-12. Qualidade dos Módulos de E/S

Conversão de Engenharia

A aba de Engineering Conversion tem por objetivo facilitar a conversão linear de escalas. É comum que determinado sinal de entrada ou saída precise ser convertido de uma escala que se refere aos limites de um sensor, por exemplo, para uma escala diferente que tenha representatividade no contexto onde esse sinal é interpretado. Com este objetivo o usuário de um sistema de controle normalmente se depara com a demanda de converter unidades de engenharia. Essa conversão geralmente está associada à relação entre pontos físicos e pontos internos.

A configuração das Conversões de Engenharia , visualizadas na Figura 4-5, segue os parâmetros descritos na Tabela 4-13. É possível a configuração de até 5120 entradas na tabela de Conversão de Engenharia.

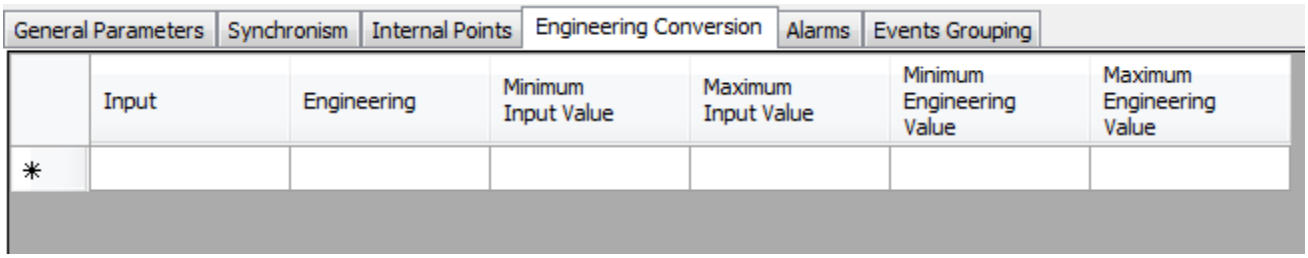


Figura 4-5. Tela de Configuração dos Parâmetros de Conversão de Engenharia

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Input	Variável simbólica de entrada da conversão.	-	Aceita variáveis simbólicas do tipo INT, DINT, UINT, UDINT, WORD e DWORD. A variável pode ser simples, array ou elemento de array e pode estar em estruturas.

Engineering	Variável simbólica que armazenará o resultado da conversão.	-	Aceita somente variáveis do tipo REAL. A variável pode ser simples, array ou elemento de array e pode estar em estruturas.
Minimum Input Value	Menor valor da entrada.	-	Constante numérica que deve obedecer aos limites do tipo da variável utilizada na "Entrada".
Maximum Input Value	Maior valor de entrada.	-	Constante numérica que deve obedecer aos limites do tipo da variável utilizada na "Entrada".
Minimum Engineering Value	Menor valor de engenharia.	-	Constante numérica que deve obedecer aos limites do tipo REAL.
Maximum Engineering Value	Maior valor de engenharia.	-	Constante numérica que deve obedecer aos limites do tipo REAL.

Tabela 4-13. Configuração dos Parâmetros de Conversão de Engenharia

Alarmes

Alarmes são variáveis que armazenam um valor binário, que podem ser do tipo verdadeiro ou falso, sendo este valor o resultado da comparação de uma variável de entrada com um dado setpoint (valor de referência). O valor de um alarme geralmente é utilizado para disparar eventos de informação de estados da aplicação ou processo.

A configuração dos Alarmes, visualizadas na Figura 4-6, segue os parâmetros descritos na Tabela 4-14. É possível a configuração de até 5120 entradas na tabela de Alarmes.

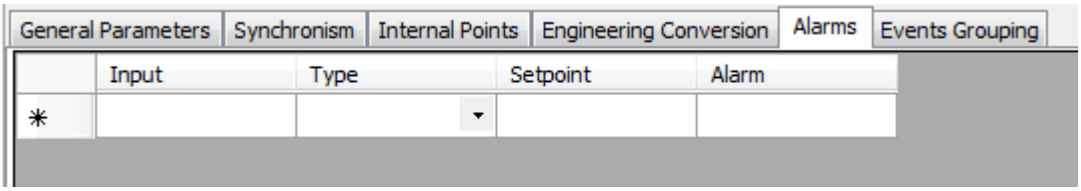


Figura 4-6. Tela de Configuração dos Alarmes

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Input	Variável simbólica que possui a informação utilizada para geração do alarme.	-	Aceita variáveis do tipo INT, DINT, LINT, UINT, UDINT, ULINT, WORD, DWORD, LWORD, REAL ou LREAL. A variável pode ser simples, array ou elemento de array e pode estar em estruturas.
Type	Tipo de comparação que será realizada entre a entrada e o setpoint.	-	Pode assumir dois valores predefinidos: "Maior que" ou "Menor que".
Setpoint	Valor de referência utilizado para identificar se o valor da entrada gerou ou não um alarme.	-	Aceita variáveis simbólicas do tipo INT, DINT, LINT, UINT, UDINT, ULINT, WORD, DWORD, LWORD, REAL ou LREAL. A variável pode ser simples, array ou elemento de array e pode estar em estruturas. São aceitas também constantes numéricas que respeitem os limites dos tipos LINT, LWORD e LREAL.
Alarm	Variável simbólica que receberá o valor VERDADEIRO ou FALSO, dependendo do resultado da comparação entre "Entrada" e "Setpoint".	-	Aceita somente variáveis do tipo BOOL. A variável pode ser simples, array ou elemento de array e pode estar em estruturas.

Tabela 4-14. Configuração dos Alarmes

Agrupamento de Eventos

O agrupamento de eventos permite resumir um grupo de eventos em um único e novo evento, usando para isso um novo ponto digital. Um agrupamento de eventos é definido por um conjunto de pontos de entrada digital, por uma variável que armazena o resultado do agrupamento e pelo tempo de atraso máximo para a geração do evento.

O funcionamento do agrupamento de eventos consiste em calcular o valor e a qualidade da variável resumo através de uma lógica utilizando os pontos de entrada do agrupamento. Esta lógica é realizada a partir dos eventos ocorridos para as entradas, os quais são armazenados em uma fila interna dedicada para esta funcionalidade. A estampa de tempo da variável de resultado será determinada em função da estampa de tempo das entradas.

Para eventos relacionados à mudança de valor, é realizada uma lógica do tipo “OU”. Desta forma o evento da transição "0" para "1" da variável resultado é datado pelo evento de "0" para "1" mais antigo das entradas do grupo, enquanto o evento da transição "1" para "0" da variável resultado é datado pelo evento de "1" para "0" mais recente das entradas. Para eventos relacionados à mudança da qualidade, é realizada uma lógica do tipo “E”, isto é, basta que apenas uma entrada mude a qualidade para um valor diferente de GOOD para que a qualidade do resultado mude. Desta forma, o evento de qualidade da variável resultado é datado sempre pelo evento de qualidade mais antigo.

Após a ocorrência de um ou mais eventos que resultem em uma alteração do valor ou qualidade da variável de resultado, é iniciada a contagem de um tempo denominado tempo de atraso máximo. Ao final desta janela de tempo, é realizado o cálculo do valor e qualidade da variável resultado através da avaliação (em ordem cronológica) de todos os eventos previamente armazenados na fila interna dedicada para esta funcionalidade. Caso um evento que tenha ocorrido durante a janela de tempo definido pelo tempo de atraso máximo chegue após este tempo ter expirado, é gerado um alarme de atraso excessivo. Este alarme indica que o resultado do agrupamento pode não ser válido, pois um ou mais eventos não foram utilizados no cálculo por terem chegado atrasados.

A fila utilizada para armazenar os eventos relacionados às entradas possui capacidade para armazenar 8192 eventos, o que é suficiente para garantir o correto funcionamento em condições normais de operação. Caso seja gerada uma grande quantidade de eventos durante o tempo de atraso máximo de forma a encher esta fila, os novos eventos serão descartados (isto é, mantém os mais antigos).

A Figura 4-7 demonstra o funcionamento do agrupamento de eventos. Neste exemplo, o resultado do agrupamento é armazenado na variável R, enquanto as entradas do grupo são compostas pelas variáveis P1, P2 e P3.

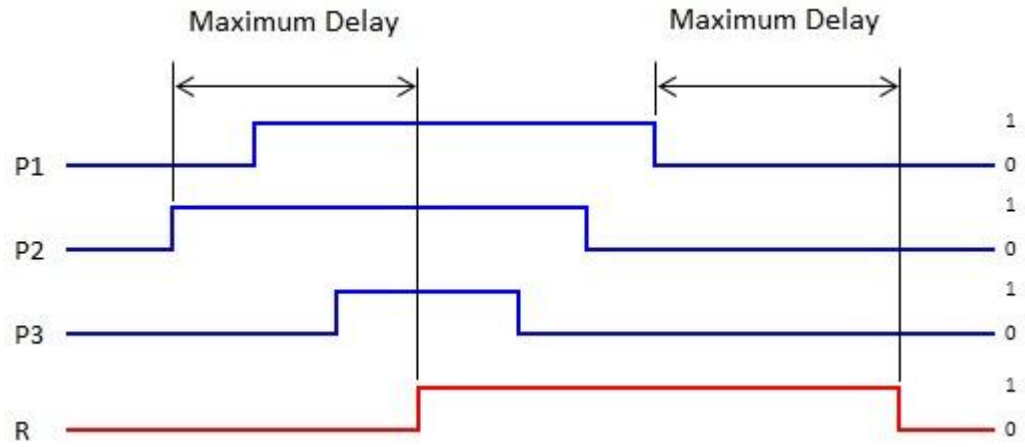


Figura 4-7. Funcionamento do Agrupamento de Eventos

ATENÇÃO:
Se um agrupamento de pontos for utilizado como entrada para outro agrupamento, é necessário se certificar de que o parâmetro "Atraso Máximo" do primeiro agrupamento seja menor do que o do grupo no qual ele está sendo inserido.

A configuração do agrupamento de eventos, visualizadas na Figura 4-8 permite a configuração de até 128 entradas na tabela de Agrupamento de Eventos. A Tabela 4-15 mostra os valores padrão e admissíveis para cada campo configurável.

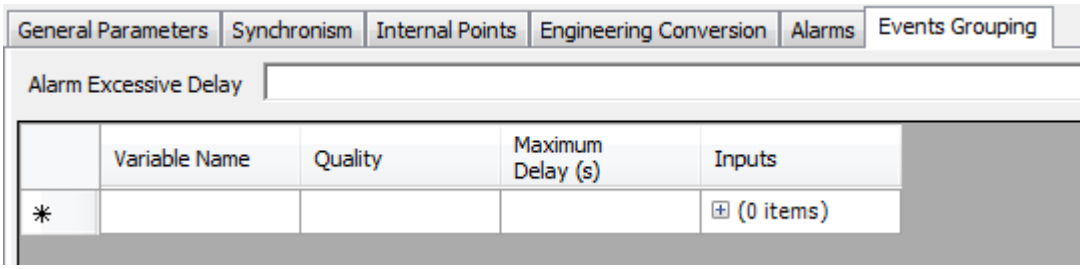


Figura 4-8. Tela de Configuração do Agrupamento de Eventos

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Alarm Excessive Delay	Nome da variável digital que indica se um evento chegou tarde demais e foi descartado na composição do resumo de eventos.	-	A variável pode ser simples ou elemento de array e pode estar em estruturas. Deve ser do tipo BOOL.
Variable Name	Armazena o resultado do agrupamento de eventos.		A variável pode ser simples ou elemento de array e pode estar em estruturas. Deve ser do tipo BOOL.
Quality	Armazena a qualidade do agrupamento de eventos em uma variável.	-	Pode ser simples ou elemento de array e pode estar em estruturas. A variável de qualidade deve ser do tipo QUALITY, disponível na biblioteca LibDataTypes.
Maximum Delay (s)	Tempo em segundos a partir da chegada do primeiro evento que será aguardado para que os demais eventos do grupo cheguem.	-	De 1 a 60 segundos.
Inputs	Configuração dos pontos digitais que compõem o grupo de eventos monitorados. Cada linha corresponde a um ponto digital que será monitorado. Um ponto digital só pode estar em um grupo.	-	A variável pode ser simples ou elemento de array e pode estar em estruturas. Deve ser do tipo BOOL.

Tabela 4-15. Configuração do Agrupamento de Eventos

Configuração das Interfaces Seriais

COM 1

A interface de comunicação COM 1 é composta por um conector tipo DB9 fêmea para o padrão RS-232C e RS-485(não isolada). Permite a comunicação ponto a ponto (ou em rede utilizando conversor) nos protocolos abertos, MODBUS RTU escravo ou MODBUS RTU mestre.

A Tabela 4-16, segue os parâmetros que devem ser configurados para o bom funcionamento da aplicação.

Quando utilizado o protocolo MODBUS Master/Slave, alguns destes parâmetros (como Modo Serial, Bits de Dado, Threshold RX e Eventos Seriais) são ajustados automaticamente pela ferramenta MasterTool para o correto funcionamento deste protocolo.

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Tipo Serial	Configuração do tipo do canal serial	RS-232C	RS-232C ou RS-485
Taxa de Transmissão	Velocidade da porta de comunicação serial	115200	600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bps
Paridade	Configura a paridade da porta serial	Sem Paridade	Ímpar Par Paridade sempre Um Paridade Sempre Zero Sem Paridade
Bits de Dado	Configura o número de bits de dados em cada caractere da comunicação serial.	8	5, 6, 7 e 8
Bits de Parada	Configura os bits de parada da porta serial	1	1, 1.5 e 2
Modo Serial	Configura o modo de operação da porta serial	Modo Normal	- Modo Estendido: Modo estendido de operação da comunicação serial, no qual são fornecidas informações sobre o frame de dados recebido. - Modo Normal: Modo normal de operação da comunicação serial

Tabela 4-16. Configurações da COM 1

Nota:

Modo Estendido: Neste modo de operação da comunicação serial são fornecidas informações sobre o frame de dados recebido. As informações disponibilizadas são as seguintes:

Um byte para o dado recebido (RX_CHAR : BYTE): Armazena os cinco, seis, sete ou oito bits dos dados recebidos, dependendo da configuração da serial.

Um byte para os erros no sinal (RX_ERROR : BYTE): Tem o seguinte formato:

- Bit 0: 0 - o caractere nos bits 0 a 7 é válido. 1 - o caractere nos bits 0 a 7 não é válido (ou pode não ser válido), devido aos problemas indicados nos bits 10 a 15
- Bit 1: Não utilizado
- Bit 2: Não utilizado
- Bit 3: Erro de interrupção na UART. A entrada serial permaneceu na lógica 0 (paridade sempre zero) por um tempo maior do que um caractere (bit de partida + bit de dados + bit de paridade + bit de parada)
- Bit 4: Erro no frame UART. A lógica 0 (paridade sempre zero) foi lida quando o primeiro bit de parada era esperado, sendo que deveria ser lógica 1 (paridade sempre um)
- Bit 5: Erro de paridade UART. O bit de paridade lido não está de acordo com o bit de paridade calculado
- Bit 6: Erro de overrun UART. Dados foram perdidos durante a leitura do FIFO UART, porque novos caracteres foram recebidos antes de os antigos serem removidos. Esse erro somente será indicado no primeiro caractere lido após a indicação do erro de overrun. Isso significa que alguns dados antigos foram perdidos
- Bit 7: Erro de overrun na fila RX: Esse caractere foi escrito quando a fila RX foi completa, sobrescrevendo caracteres não lidos

Dois bytes para o sinal de estampa de tempo (RX_TIMESTAMP : WORD): Indica o tempo de silêncio, no intervalo de 0 a 65535, usando como base de tempo 10 us. Satura em 655,35 ms, caso o tempo de silêncio seja maior do que 65535 unidades. O RX_TIMESTAMP de um caractere mede o tempo desde uma referência que pode ser uma das três opções abaixo:

- Na maioria dos casos, o final do caractere anterior
- Configuração da porta serial
- O fim da transmissão serial usando o SERIAL_TX FB, ou seja, quando o último caractere foi enviado na linha

Além de medir o tempo de silêncio antes de cada caractere, o RX_TIMESTAMP também é importante, pois mede o tempo de silêncio do último caractere da fila RX.

A medição do silêncio é importante para implementar corretamente alguns protocolos, como MODBUS RTU. Esse protocolo especifica um inter-frame maior do que 3,5 caracteres e um inter-byte menor do que 1,5 caracteres.

Bits de Dado: A configuração Bits de Dado das interfaces seriais limita os campos Bits de Parada e Paridade da comunicação, ou seja, o número de bits de parada e o método de paridade irão variar de acordo com o número de bits de dado. A Tabela 4-10 mostra as configurações permitidas para a interfaces COM 1.

Bits de Dados	Bits de Parada	Paridade
5	1,1.5	SEM PARIDADE, ÌMPAR, PAR, PARIDADE SEMPRE UM, PARIDADE SEMPRE ZERO
6	1, 2	SEM PARIDADE, ÌMPAR, PAR, PARIDADE SEMPRE UM, PARIDADE SEMPRE ZERO
7	1, 2	SEM PARIDADE, ÌMPAR, PAR, PARIDADE SEMPRE UM, PARIDADE SEMPRE ZERO
8	1, 2	SEM PARIDADE, ÌMPAR, PAR, PARIDADE SEMPRE UM, PARIDADE SEMPRE ZERO

Tabela 4-17. Configurações Específicas

Configurações Avançadas

As configurações avançadas são relacionadas aos sinais de controle da comunicação serial, ou seja, quando se faz necessária a utilização de um controle mais apurado da transmissão e recepção dos dados.

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Parâmetros de Porta Avançados			
Sinais de modem	Realiza o controle de requisição para transmissão de um comando, através da interface RS-232C.	RTS Sempre Desligado	- RTS: Habilitado no início da transmissão e reiniciado, o mais rápido possível, após o final da transmissão. Ex: Controle de um conversor RS-232C/RS-485 externo. - RTS Sempre Desligado: Sempre desabilitado - RTS Sempre Ligado: Sempre habilitado - RTS/CTS: Caso o CTS esteja desabilitado, o RTS é habilitado. Então, aguarda-se o CTS ser habilitado para a transmissão começar e o RTS é reiniciado, o mais rápido possível, no final da transmissão. Ex: Controle de rádio modems com o mesmo sinal de modem - RTS Manual: o usuário é responsável por controlar todos os sinais de controle

Threshold de RX da UART	Quantidade de bytes que devem ser recebidos para gerar uma nova interrupção na UART. Valores baixos fazem o TIMESTAMP mais preciso quando o MODO ESTENDIDO é utilizado e minimiza os erros de overrun. No entanto, valores baixos podem causar muitas interrupções que podem retardar a UCP.	8	1, 4, 8 e 14
Eventos Seriais			
RX em TX	Se habilitado todos os bytes recebidos durante a transmissão serão descartados em vez de ir para a fila de RX. Utilizado para desabilitar a operação full-duplex na interface RS-232C.	Desabilitado	- Habilitado: Configuração habilitada - Desabilitado: Configuração desabilitada
Evento RX DCD	Se habilitado gera um evento externo devido à mudança do sinal DCD.	Habilitado	- Habilitado: Configuração habilitada - Desabilitado: Configuração desabilitada
Evento RX CTS	Se habilitado gera um evento externo devido à mudança do sinal CTS.	Habilitado	- Habilitado: Configuração habilitada - Desabilitado: Configuração desabilitada

Tabela 4-18. Configurações Avançadas COM 1

Notas:

RX em TX: Este parâmetro avançado é valido para configurações RS-232C e para RS-422.

Evento RX DCD: Os eventos externos, como o sinal de DCD da COM 1, só podem ser associados às tarefas do perfil de projeto personalizado, descrito na seção Perfil Personalizado do capítulo de Perfis de Projeto.

Evento RX CTS: Os eventos externos, como o sinal de CTS da COM 1, só podem ser associados às tarefas do perfil de projeto personalizado, descrito na seção Perfil Personalizado do capítulo de Perfis de Projeto.

COM 2

A interface de comunicação COM 2 é composta por um conector tipo DB9 fêmea para os padrões RS-422C e RS-485. Permite a comunicação ponto a ponto ou em rede nos protocolos abertos, MODBUS RTU escravo ou MODBUS RTU mestre.

A Tabela 4-19 segue os parâmetros que devem ser configurados para o bom funcionamento da aplicação.

Quando utilizado o protocolo MODBUS Master/Slave, alguns destes parâmetros (como Modo Serial, Bits de Dado, Threshold RX e Eventos Seriais) são ajustados automaticamente pela ferramenta MasterTool para o correto funcionamento deste protocolo.

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Tipo Serial	Configuração do tipo do canal serial	RS-485	RS-422 ou RS-485
Taxa de Transmissão	Velocidade da porta de comunicação serial	115200	600, 1200, 1800, 2400, 4800, 9600, 19200,

			38400, 57600, 115200 bps
Paridade	Configura a paridade da porta serial	Sem Paridade	Ímpar Par Paridade Sempre Um Paridade Sempre Zero Sem Paridade
Bits de Dado	Configura o número de bits de dados em cada caractere da comunicação serial.	8	5, 6, 7 e 8
Bits de Parada	Configura os bits de parada da porta serial	1	1, 1.5 e 2
Modo Serial	Configura o modo de operação da porta serial	Modo Normal	- Modo Estendido: Modo estendido de operação da comunicação serial, no qual são fornecidas informações sobre o frame de dados recebido (ver nota seção COM 1). - Modo Normal: Modo normal de operação da comunicação serial

Tabela 4-19. Configuração COM 2

A configuração Bits de Dado das interfaces seriais limita os campos Bits de Parada e Paridade da comunicação, ou seja, o número de bits de parada e o método de paridade irá variar de acordo com o número de bits de dado. A Tabela 4-20 mostra as configurações permitidas para a interface COM 2.

Bits de Dados	Bits de Parada	Paridade
5	1, 1.5	SEM PARIDADE, ÌMPAR, PAR, PARIDADE SEMPRE UM, PARIDADE SEMPRE ZERO
6	1, 2	SEM PARIDADE, ÌMPAR, PAR, PARIDADE SEMPRE UM, PARIDADE SEMPRE ZERO
7	1, 2	SEM PARIDADE, ÌMPAR, PAR, PARIDADE SEMPRE UM, PARIDADE SEMPRE ZERO
8	1, 2	SEM PARIDADE, ÌMPAR, PAR, PARIDADE SEMPRE UM, PARIDADE SEMPRE ZERO

Tabela 4-20. Configurações Específicas

Configurações Avançadas

As configurações avançadas são relacionadas aos sinais de controle da comunicação serial, ou seja, quando se faz necessária a utilização de um controle mais apurado da transmissão e recepção dos dados.

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Threshold de RX da UART	Quantidade de bytes que devem ser recebidos para gerar uma nova interrupção na UART. Valores baixos fazem o TIMESTAMP mais preciso quando o MODO ESTENDIDO é utilizado e minimiza os erros de overrun. No entanto, valores baixos podem causar muitas interrupções que podem retardar a UCP.	8	1, 4, 8 e 14

Tabela 4-21. Configurações Avançadas COM 2

Configuração das Interfaces Ethernet

A UCP Hadron Xtorm possui seis interfaces Ethernet locais denominadas NET 1 à NET 6, as quais podem operar de forma independente ou sob a forma de pares. Cada interface (ou cada par) deve ser configurada em uma subrede distinta.

Interfaces Ethernet Locais

NET 1 à NET 6

As interfaces NET 1 à NET 6 são compostas por um conector tipo RJ45 de comunicação no padrão 10/100Base-TX. Permitem a comunicação ponto a ponto ou em rede nos protocolos suportados. A NET 1 está sempre habilitada, e é através dela (ou do par NET 1+2) que é realizada a comunicação com a ferramenta MasterTool Xtorm. Já as interfaces NET 2 à NET6 podem ser habilitadas/desabilitadas através da configuração.

A Tabela 4-22 descreve os parâmetros de configuração para as interfaces Ethernet:

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Habilitar	Habilita a interface Ethernet (somente para NET 2 à NET 6)	Desabilitado	Desabilitado Habilitado
Endereço IP	Endereço IP do Controlador no barramento Ethernet.	192.168.15.1 (NET 1) 192.168.16.1 (NET 2) 192.168.17.1 (NET 3) 192.168.18.1 (NET 4) 192.168.19.1 (NET 5) 192.168.20.1 (NET 6)	1.0.0.1 a 223.255.255.254
Máscara de Subrede	Máscara de subrede do Controlador no barramento Ethernet.	255.255.255.0	0.0.0.1 a 255.255.255.254
Endereço do Gateway	Endereço do Gateway do Controlador no barramento Ethernet.	192.168.15.253 (NET 1) 192.168.16.253 (NET 2) 192.168.17.253 (NET 3) 192.168.18.253 (NET 4) 192.168.19.253 (NET 5) 192.168.20.253 (NET 6)	1.0.0.1 a 223.255.255.254

Tabela 4-22. Configurações das interfaces NET 1 à NET6

ATENÇÃO:
Não é possível configurar duas ou mais interfaces Ethernet de uma UCP na mesma subrede, sendo este tipo de configuração bloqueada pela ferramenta MasterTool Xtorm. Desta forma, cada interface Ethernet deve ser configurada em uma subrede distinta.

Portas TCP Reservadas

As seguintes portas TCP das interfaces Ethernet, tanto locais quanto remotas, são utilizadas por serviços da UCP e, portanto, são reservadas e não devem ser utilizadas pelo usuário: 80, 8080, 1217, 1740, 1741, 1742,1743 e 11740.

Configurações Avançadas das Interfaces Ethernet

Os canais Ethernet da UCP HX3040 podem ser configurados em três modos distintos de funcionamento: de forma individual, ou organizados em pares NIC Teaming ou modo Switch. Os pares possíveis são NET 1+2, NET 3+4 e NET 5+6, sendo que a configuração do par fica armazenada sempre na NET ímpar. Desta forma, quando configuradas como parte de um par, os campos de configuração das NETs 2, 4 e 6 ficam desabilitados.

As configurações avançadas das interfaces Ethernet, visualizadas na Figura 4-9, seguem os parâmetros descritos na Tabela 4-23:

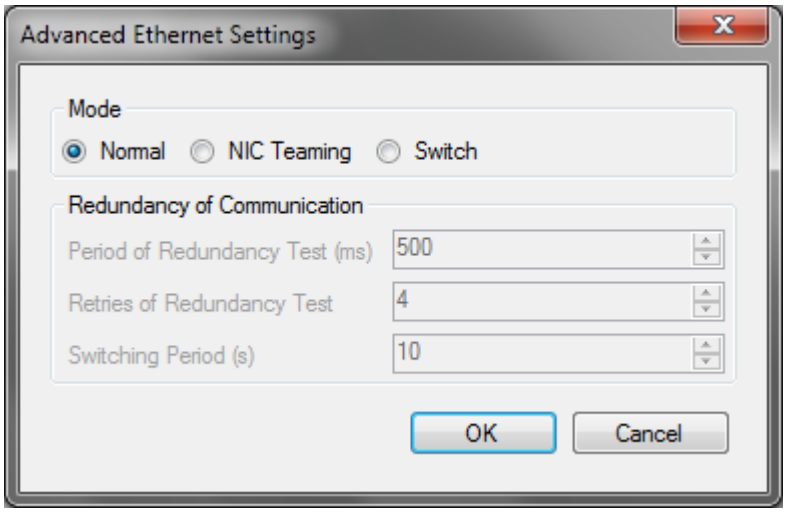


Figura 4-9. Tela de Configurações Avançadas das Interfaces Ethernet

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Mode	Habilita o modo de funcionamento da Interface Ethernet	Normal	Normal Nic Teaming Switch

Tabela 4-23. Configurações Avançadas das Interfaces Ethernet

Modo Normal: Neste modo, a interface opera como uma porta Ethernet independente, não havendo qualquer relação com a interface subsequente.

Modo NIC Teaming: Neste modo, a interface forma um par redundante com a interface subsequente, operando em um esquema do tipo ativa/reserva. Um par NIC Teaming possui um único endereço IP, associado à interface que estiver ativa no momento. Desta forma, um cliente como um SCADA ou Mastertool Xtorm, conectado a um servidor na UCP, não precisa se preocupar em trocar o endereço IP caso haja falha em alguma das portas do par. Além disso, cada uma das interfaces que formam um par NIC Teaming possuem diagnósticos separados, com o intuito de facilitar a depuração de falhas que eventualmente possuam surgir.

Quando o Modo NIC Teaming for selecionado , na mesma tela automaticamente outros parâmetros serão habilitados e deverão ser configurados:

- Período de Teste da Redundância (ms): Período de envio do frame de teste de comunicação entre as duas NETs. Pode ser configurado com valores entre 100 e 9900, padrão 500
- Número de Retentativas do Teste da Redundância: Numero máximo de vezes que a NET que enviou o frame irá esperar pela resposta. Pode ser configurado com valores entre 1 e 100, padrão 4
- Período para Chaveamento (s): Tempo máximo que a NET Ativa irá esperar por um pacote qualquer. Pode ser configurado com valores entre 1 e 25, padrão 10

Caso o tempo de resposta do Teste da Redundância atinja Período de Teste vezes o Número de Retentativas e a interface ativa permaneça por um tempo maior que o Período para Chaveamento sem receber nenhum pacote, ocorrerá um switchover, tornando ativa a interface que até então estava inativa. É importante ressaltar que há um delay entre a detecção da falha e ativação da interface inativa devido ao tempo necessário para sua configuração. Tal delay pode chegar a algumas dezenas de milissegundos

Quando uma das NETs estiver ativa, esta assumirá o endereço de IP configurado, permanecendo a NET inativa com seus parâmetros de Endereço de IP, Máscara de Subrede e Endereço do Gateway em branco nos diagnósticos da UCP.

ATENÇÃO:
Quando um comando de Reset Origem for executado em uma UCP configurada com NIC Teaming nas interfaces frontais NETx, apenas a interface que estava ativa antes do comando continuará acessível. Após o comando, a interface acessível poderá ser visualizada no Menu Informativo e de Configuração da UCP.

Modo Switch: Neste modo, a interface forma um par com a interface subsequente operando como um switch Ethernet, permitindo então a comunicação por ambas às portas. Desta forma, este modo permite a ligação em "cascata" de diversas UTRs Xtorm, possibilitando a implementação de uma topologia de rede em anel. Contudo, para realizar o fechamento deste anel, é obrigatório que as extremidades desta "cascata" sejam conectadas a um switch externo que possua suporte aos protocolos STP ou RSTP. A vantagem desta arquitetura é a redução do número de portas do switch externo quando comparada à topologia de rede em estrela. Contudo, a principal desvantagem está no tempo para recuperação da rede em caso de uma falha (tempo de convergência), o qual pode ser de até 50 segundos utilizando STP ou de até 6 segundos utilizando RSTP. Desta forma, é importante avaliar se o sistema que está sendo desenvolvido pode conviver sem maiores problemas com estes tempos de interrupção da rede. Caso seja um sistema crítico que não possa conviver com interrupções tão longas (por exemplo, sistemas de controle ou proteção com tempos de resposta muito rápidos), deve ser utilizada a topologia de rede em estrela.

Pontos Duplos

A representação de Pontos digitais duplos de entrada e saída é realizada através de um tipo de dados especial chamado DBP (definido na biblioteca LibDataTypes). Este tipo consiste basicamente em uma estrutura de dois elementos do tipo BOOL chamados OFF e ON (equivalentes a TRIP e CLOSE respectivamente). Variáveis deste tipo podem ser associadas à módulos de Entrada e Saída que possuem suporte a eventos e comandos pulsados (HX1120 e HX2320 por exemplo), onde cada um dos elementos OFF e ON deve ser mapeado individualmente para cada entrada/saída física do módulo. No caso dos módulos de entrada, a memória %I é atualizada normalmente mesmo para os pontos que estiverem configurados como DBP. Já no caso dos módulos de saída, os pontos configurados como DBP não consideram mais o valor presente na memória %Q.

Maiores informações sobre a configuração de pontos duplos nos módulos de entrada e saída podem ser obtidos nos seus respectivos documentos de Características Técnicas.

Configuração de Protocolos

A Tabela 4-24 a seguir, informa os limites máximos de configuração relacionados às interfaces de comunicação e aos protocolos de comunicação da UCP. Esta tabela não se aplica ao protocolo IEC 61850, o qual possui limites específicos que estão descritos na seção IEC61850 Servidor.

	HX3040
Instâncias por Interface Serial (COMs)	1
Instâncias por Interface Ethernet (NETs)	4
Instâncias por UCP	12
Centros de Controle por UCP	4
Mapeamentos por UCP	5.000
Número de pontos mapeados	20.000
Número máximo de conexões DNP3	100

Tabela 4-24. Limites dos Protocolos por Interface e por UCP

Notas:

Instâncias por Interface Serial (COMs): Devido as suas características, cada interface serial suporta apenas uma instância de protocolo de comunicação. São exemplos de instâncias compatíveis com interfaces Seriais: MODBUS RTU Mestre e MODBUS RTU Escravo.

Instâncias por Interface Ethernet (NETs): Interfaces Ethernet suportam mais de uma instância de protocolo de comunicação, conforme informado na tabela. São exemplos de instâncias compatíveis com interfaces Ethernet: MODBUS Cliente, MODBUS Servidor, Cliente DNP3, entre outros.

Instâncias por UCP: A soma das instâncias de protocolos de comunicação, declaradas nas interfaces Seriais e Ethernet, não pode ultrapassar a quantidade máxima de instâncias suportadas pela UCP.

Centros de Controle: "Centro de Controle" é todo dispositivo cliente conectado à UCP através dos protocolos DNP3 e MMS (IEC 61850). Este campo informa a quantidade máxima de dispositivos clientes, do tipo centro de controle, suportada pela UCP. Corresponde a soma de todos os dispositivos clientes das instâncias de protocolos de comunicação Servidor DNP3 e MMS Servidor (não inclui os mestres e clientes dos protocolos MODBUS RTU Escravo e MODBUS Servidor).

Mapeamentos por UCP: Um "mapeamento" é uma relação entre uma variável interna de aplicação e um objeto do protocolo de aplicação. Este campo informa a quantidade máxima de mapeamentos suportada pela UCP. Corresponde a soma de todos os mapeamentos realizados nas instâncias de protocolos de comunicação e seus respectivos dispositivos (ver próxima tabela).

Número de pontos mapeados: É o número máximo de pontos mapeados que a UCP suporta. Lembrando que cada mapeamento suporta 1 ou mais pontos mapeados.

Número máximo de conexões DNP3: É a soma do número máximo de conexões Cliente DNP3 e Servidor DNP3 de uma UCP HX3040, não podendo exceder o número de 100 conexões no total para um funcionamento correto.

A Tabela 4-25 a seguir, informa os limites máximos de configuração e operação dos protocolos de comunicação MODBUS e DNP3, das interfaces Seriais e Ethernet.

	MODBUS RTU Mestre	MODBUS RTU Escravo	MODBUS Cliente	MODBUS Servidor	Cliente DNP3	Servidor DNP3
Quantidade de dispositivos por instância	64	1([1])	64	64([2])	32	4([3])
Quantidade de requisições por dispositivo	32	NA	32	NA	32	NA
Quantidade de requisições simultâneas por instância	1	1	128	64	32	4
Quantidade de requisições simultâneas por dispositivo	1	1	8	64	1	1

Tabela 4-25. Limites dos Protocolos de Comunicação MODBUS e DNP3

Notas:

Quantidade de dispositivos por instância:

Protocolos Mestre ou Cliente: quantidade de dispositivos escravos ou servidores suportados por cada instância de protocolo Mestre ou Cliente.

Protocolo MODBUS RTU Escravo: o limite([1]) informado diz respeito às interfaces seriais, que não permitem que um Escravo possa estabelecer comunicação, simultaneamente, através da mesma interface serial, com mais de um dispositivo Mestre. Não é necessário, e nem é possível, declarar nem configurar o dispositivo Mestre sob a instância do protocolo MODBUS RTU Escravo. O dispositivo Mestre terá acesso a todos os mapeamentos feitos diretamente na instância do protocolo MODBUS RTU Escravo.

Protocolo MODBUS Servidor: o limite([2]) informado diz respeito às interfaces Ethernet, que limitam a quantidade de conexões que podem ser estabelecidas com outros dispositivos através de uma mesma interface Ethernet. Não é necessário, e nem é possível, declarar nem configurar os dispositivos Clientes sob a instância do protocolo MODBUS Servidor. Todos os dispositivos Clientes terão acesso a todos os mapeamentos feitos diretamente na instância do protocolo MODBUS Servidor.

Protocolo Servidor DNP3: quantidade de dispositivos clientes, do tipo centro de controle, suportados por cada instância de protocolo Servidor DNP3. O limite([3]) informado pode ser menor em função dos limites totais da UCP (consultar a Tabela 4-25).

Quantidade de requisições por dispositivo: Quantidade de requisições, como por exemplo de leitura ou escrita de holding registers, que podem ser configuradas para cada um dos dispositivos (escravos ou servidores) de instâncias de protocolos Mestre ou Cliente. Este parâmetro não é aplicável (NA) às instâncias de protocolos Escravo ou Servidor.

Quantidade de requisições simultâneas por instância: Quantidade de requisições que podem ser transmitidas simultaneamente por cada instância de protocolo Cliente, ou que podem ser recebidas simultaneamente por cada instância de protocolo Servidor. Instâncias de protocolo MODBUS RTU, Mestre ou Escravo, não suportam requisições simultâneas.

Quantidade de requisições simultâneas por dispositivo: Quantidade de requisições que podem ser transmitidas simultaneamente para cada dispositivo MODBUS Servidor, ou que podem ser recebidas simultaneamente de cada dispositivo MODBUS Cliente. Dispositivos MODBUS RTU, Mestre ou Escravo, bem como dispositivos DNP3, Cliente ou Servidor, não suportam requisições simultâneas.

Comportamento dos Protocolos x Estados da UCP

A Tabela 4-26 mostra detalhadamente cada comportamento assumido para cada protocolo de comunicação existente na UCP HX3040 em todos seus estados de funcionamento, uma legenda especificando maiores informações é apresentada na Tabela 4-26:

Protocolo	Tipo	Estado operacional da UCP					
		STOP			RUN		
		Depois do download antes da aplicação iniciar	Depois de a aplicação ir para STOP (PAUSE)	Depois de uma exceção	Não redundante ou Ativo	Redundante em Reserva	Após um breakpoint na MainPrg
MODBUS	Slave/Server	×	×	×	✓	×	×
	Master/Client	×	×	×	✓	×	×
DNP3	Outstation	×	×	×	✓	×	✓
	Master	×	×	×	✓	×	✓
MMS	Server	×	×	×	✓	×	✓
GOOSE	Publisher	×	×	×	✓	×	✓
	Subscriber	×	×	×	✓	×	✓
SNTP	Client	✓	✓	✓	✓	✓	✓
HTTP	Server	✓	✓	✓	✓	✓	✓
IRIG-B	Transmitter	✓	✓	✓	✓	✓	✓
	Receiver	✓	✓	✓	✓	✓	✓

Tabela 4-26. Comportamento dos Protocolos x Estados da UCP

Legenda:

✓ - O protocolo permanece ativo e operando normalmente

× - O protocolo é desabilitado

Fila de Eventos da UCP

A UCP possui uma fila de eventos do tipo FIFO (First In, First Out) utilizada para armazenar temporariamente os eventos relacionados aos pontos de comunicação até que eles sejam transferidos ao seu destino final.

Todos os eventos de pontos de comunicação gerados na UTR são direcionados e armazenados na fila da UCP. Esta fila possui as seguintes características:

- Tamanho: 4.500 eventos
- Política de estouro: mantém os mais recentes

Esta fila de eventos é armazenada em uma área de memória retentiva (não volátil) exclusiva para este fim, e não ocupa a área de variáveis simbólicas retentivas/persistentes. Desta forma, os eventos presentes na fila da UCP e que ainda não foram transmitidos ao centro de controle não serão perdidos em um eventual desligamento da UTR.

A fila também é redundante, isto é, é sincronizada ciclo a ciclo entre as duas UCPs quando se utiliza redundância de UCP. Maiores informações podem ser encontradas na seção específica sobre redundância de UCP.

A entrada e saída de eventos nesta fila segue um conceito de produtor/consumidor. Produtores são aqueles elementos do sistema capazes de gerar eventos, adicionando eventos na fila da UCP, enquanto os consumidores são os elementos do sistema que recebem e utilizam estes eventos, retirando eventos da fila da UCP. A Figura 4-10 descreve este funcionamento, incluindo exemplo de alguns consumidores e produtores de eventos:

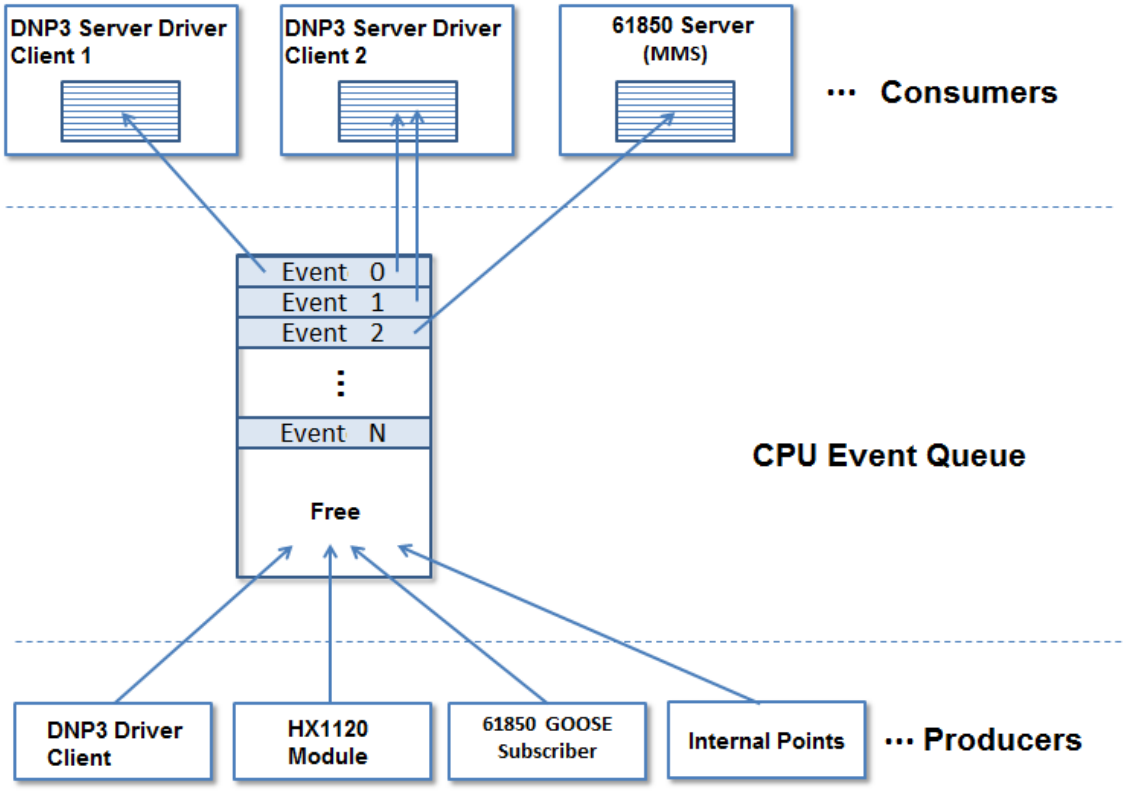


Figura 4-10. Fila de Eventos da UCP

Consumers

Os consumidores são tipicamente drivers de comunicação que irão se comunicar com um SCADA ou IHM. Após serem armazenados na fila da UCP, os consumidores recebem os eventos relativos a pontos de comunicação mapeados na sua configuração. Estes eventos são então armazenados em uma fila de eventos própria do consumidor, cujo tamanho e funcionamento é descrito na seção específica do driver.

Princípios de Funcionamento da Fila

Uma vez armazenado na fila da UCP, cada evento é transmitido para o consumidor que possuir este ponto de comunicação na sua base de dados. Na figura acima, o Evento 0 é referente a um ponto de comunicação mapeado para 2 centros de controle DNP3 (Cliente 1 e 2). Já o Evento 1 é referente a um ponto de comunicação mapeado apenas para um centro de controle DNP3 (Cliente 2). Por sua vez, o Evento 2 é referente a um ponto de comunicação mapeado para um centro de controle 61850.

Os eventos permanecem armazenados na fila da UCP até que todos os seus consumidores confirmem a sua recepção. O critério utilizado para confirmar a recepção é específico de cada consumidor. No caso específico do DNP3 Servidor, a confirmação ocorre apenas quando o centro de controle (SCADA) confirmar a recepção da mensagem (o que oferece uma boa segurança contra perda de eventos em caso de falhas de rede). Já para outros tipos de consumidores (como é o caso dos Buffered Reports no 61850 Server), a confirmação ocorre quando o evento foi transferido para a fila interna do driver. A ocupação da fila de eventos da UCP pode ser monitorada através dos diagnósticos disponibilizados na área de diagnósticos da UCP através da variável DG_HX3040.tDetailed.Events.*, a qual fornece também informações sobre estouro da fila, entre outras.

Producers

Os produtores são tipicamente drivers de comunicação ou elementos internos da UTR que são capazes de gerar eventos. A figura acima mostra alguns exemplos:

- Driver DNP3 Cliente: os eventos gerados e armazenados nos outstations, e então são lidos pelo driver e inseridos na fila da UCP
- Módulo HX1120: os eventos são gerados e armazenados internamente no módulo, e então são lidos pelo barramento local da UTR e inseridos na fila da UCP
- 61850 GOOSE Subscriber: os eventos são gerados no momento da recepção de uma mensagem GOOSE (desde que os dados recebidos causem uma alteração de valor em alguma variável), e então são inseridos na fila da UCP
- Pontos Internos: este é um elemento interno no firmware da UTR, o qual realiza a detecção de eventos a cada ciclo de execução (MainTask) para aqueles pontos de comunicação que não possuem uma origem definida e então insere os eventos na fila da UCP. O número máximo de eventos que podem ser detectados a cada ciclo da MainTask é igual ao tamanho da fila de eventos da UCP. Caso seja gerado um número de eventos maior que este em um mesmo ciclo da MainTask, os eventos excedentes serão perdidos
- Driver MODBUS (Cliente/Servidor/Mestre/Escravo): as alterações de valor em variáveis causadas por leituras/escritas MODBUS são detectadas a cada ciclo da MainTask e então os eventos são inseridos na fila da UCP. No caso dos drivers Cliente/Mestre, são gerados também eventos de qualidade quando houver falha de comunicação com o dispositivo escravo

Notas:

- HX1120: O módulo HX1120 armazena internamente os eventos em uma estrutura que pode conter de 1 a 32 eventos. Quando apenas 1 entrada muda de estado, esta estrutura armazenará somente 1 evento. Agora, quando todas as 32 entradas mudarem de estado simultaneamente, esta estrutura armazenará 32 eventos. O módulo possui uma fila interna com capacidade para armazenar até 420 estruturas (isto é, de 420 a 13.440 eventos). A UCP realiza através do barramento local a leitura de 38 estruturas a cada 3 ciclos de execução da UTR (MainTask).

Interceptação de Comandos Oriundos do Centro de Controle

A UTR Xtorm possui um bloco de função que permite que comandos de seleção e operação para pontos de saída recebidos por drivers servidores (DNP3 Server, 61850 Server, etc...) sejam tratados pela lógica de usuário. Este recurso permite a implementação de intertravamentos, bem como a manipulação dos dados do comando recebido na lógica do usuário ou ainda o redirecionamento do comando para diferentes IEDs.

A interceptação de comandos é implementada pelo bloco de função CommandReceiver. Os parâmetros de entrada e saída são descritos nas tabelas a seguir:

Parâmetro	Tipo	Descrição
bExec	BOOL	Quando TRUE, executa o comando
bDone	BOOL	Indica que os dados de saída do comando já foram processados, liberado o bloco de função para recepção de um novo comando.
dwVariableAddr	DWORD	Endereço da variável que receberá o comando
eCommandResult	ENUM	Ação de entrada definida pelo usuário a partir da seguinte lista: SUCCESS(0), NOT_SUPPORTED(1) BLOCKED_BY_SWITCHING_HIERARCHY(2) SELECT_FAILED(3) INVALID_POSITION(4) POSITION_REACHED(5) PARAMETER_CHANGE_IN_EXECUTION(6) STEP_LIMIT(7) BLOCKED_BY_MODE(8) BLOCKED_BY_PROCESS(8) BLOCKED_BY_INTERLOCKING(10) BLOCKED_BY_SYNNCHROCHECK(11) COMMAND_ALREADY_IN_EXECUTION(12) BLOCKED_BY_HEALTH (13) ONE_OF_N_CONTROL(14) ABORTION_BY_CANCEL(15) TIME_LIMIT_OVER(16) ABORTION_BY_TRIP(17) OBJECT_NOT_SELECTED(18) OBJECT_ALREADY_SELECTED(19) NO_ACCESS_AUTHORITY(20) ENDED_WITH_OVERSHOOT(21) ABORTION_DUE_TO_DEVIATION(22) ABORTION_BY_COMMUNICATION_LOSS(23) BLOCKED_BY_COMAND(24) NONE(25) INCONSISTENT_PARAMETERS(26) LOCKED_BY_OTHER_CLIENT(27) HARDWARE_ERROR(29) UNKNOWN(30)
dwTimeout	DWORD	Time-out definido para o usuário, é um parâmetro de entrada para a função

Tabela 4-27. Parâmetros de Entrada do Bloco de Função CommandReceiver

Parâmetro	Tipo	Descrição
bCommandAvailable	BOOL	Indica que foi interceptado um comando, e que os dados estão disponíveis para serem processados
sCommand	STRUCT	Esta estrutura armazena os dados do comando recebido, a qual é composta pelo seguintes campos: eCommand sSelectParameters sOperateParameters sCancelParameters *A descrição de cada campo segue nesta seção

eStatus	ENUM	Ação de saída da função a partir do resultado obtido, conforme lista: OK_SUCCESS(0) ERROR_FAILED(1) ERROR_NOTSUPPORTED(2) ERROR_PARAMETER(3) ERROR_MODULE_ABSENT(4) ERROR_MODULE_NOTCONFIGURED(5) ERROR_MODULE_NOTRUNNING(6) ERROR_MODULE_COMMFAIL(7) ERROR_MODULE_NOTFOUND(8)

Tabela 4-28. Parâmetros de saída do Bloco de Função CommandReceiver

Os comandos suportados são descritos na Tabela 4-29:

Parâmetro	Tipo	Descrição
eCommand	ENUM	NO_COMMAND SELECT OPERATE

Tabela 4-29. Comandos Suportados do Bloco de Função CommandReceiver

Os parâmetros que compõem as estruturas sSelectParameters, sOperateParameters e sCancelParameters são descritos nas tabelas a seguir:

Parâmetro	Tipo	Descrição
sSelectConfig	STRUCT	Configuração do comando de seleção recebido. Os parâmetros desta estrutura estão descritos na Tabela 4-110
sValue	STRUCT	Valor recebido em um select, quando recebido um comando de seleção com valor. Os parâmetros desta estrutura estão descritos na Tabela 4-34

Tabela 4-30. Parâmetros sSelectParameters

Parâmetro	Tipo	Descrição
bSelectWithValue	**BOOL**	Quando true indica a recepção de um comando de seleção com valor.

Tabela 4-31. Parâmetros sSelectConfig

Parâmetro	Tipo	Descrição
sOperateConfig	STRUCT	Configuração do comando de seleção recebido. Os parâmetros desta estrutura estão descritos na Tabela 4-33. Parâmetros sOperateConfig
sValue	STRUCT	Campo de valor referente ao comando de operação recebido. Os parâmetros desta estrutura estão descritos na Tabela 4-34

Tabela 4-32. Parâmetros sOperateParameters

Parâmetro	Tipo	Descrição
bDirectOperate	BOOL	Quando true indica que foi recebido um comando de operação sem select.
bNoAcknowledgement	BOOL	Quando true indica que foi recebido um comando que não requer a confirmação de recebimento.
bTimedOperate	BOOL	Quando true indica que foi recebido um comando de operação ativado por tempo.
liOperateTime	LINT	Programação do instante em que deve ser executado o comando. Este campo é válido somente quando bTimedOperate for true.

bTest	BOOL	Quando true indica que o comando recebido foi enviado somente para fins de teste, desta forma o mesmo não deve ser executado.

Tabela 4-33. Parâmetros sOperateConfig

Parâmetro	Tipo	Descrição
eParamType	ENUM	Informa o tipo de commando que foi recebido: NO_COMMAND SINGLE_POINT_COMMAND DOUBLE_POINT_COMMAND INTEGER_STATUS_COMMAND ENUMERATED_STATUS_COMMAND ANALOGUE_VALUE_COMMAND
sSinglePoint	STRUCT	Quando recebido um comando, em função do tipo do comando recebido, definido por eParamType, é preenchida a estrutura de dados correspondente. Os parâmetros destas estruturas são apresentadas na Tabela 4-35 e Tabela 4-39
sDoublePoint	STRUCT	
sIntegerStatus	STRUCT	
sEnumeratedStatus	STRUCT	
sAnalogueValue	STRUCT	

Tabela 4-34. Parâmetros sValue

Parâmetro	Tipo	Descrição
bValue	BOOL	Valor de operação do ponto.
sPulseConfig	STRUCT	Os parâmetros de configuração de um comando pulsado são armazenados nesta estrutura. Os parâmetros desta estrutura são apresentados na Tabela 4-40

Tabela 4-35. Parâmetros sSinglePoint

Parâmetro	Tipo	Descrição
bValue	BOOL	Valor de operação do ponto.
sPulseConfig	STRUCT	Os parâmetros de configuração de um comando pulsado são armazenados nesta estrutura. Os parâmetros desta estrutura são apresentados na Tabela 4-40

Tabela 4-36. Parâmetros sDoublePoint

Parâmetro	Tipo	Descrição
diValue	DINT	Valor de operação do ponto.

Tabela 4-37. Parâmetros sIntegerStatus

Parâmetro	Tipo	Descrição
dwValue	DWORD	Valor de operação do ponto.

Tabela 4-38. Parâmetros sEnumeratedStatus

Parâmetro	Tipo	Descrição
eType	ENUM	Informa o tipo de dado do valor analógico recebido: INTEGER FLOAT
diValue	DINT	Valor de operação do ponto, no formato inteiro
fValue	DINT	Valor de operação do ponto, no formato float.

Tabela 4-39. Parâmetros sAnalogueValue

Parâmetro	Tipo	Descrição
bPulseCommand	BOOL	Quando true indica que o comando recebido é pulsado.
dwOnDuration	DWORD	Esta é a duração, expressa em milissegundos, que a saída digital deve permanecer ligada.
dwOffDuration	DWORD	Esta é a duração, expressa em milissegundos, que a saída digital deve permanecer desligada.
dwPulseCount	DWORD	Número de vezes que o comando deve ser executado.

Tabela 4-40. Parâmetros sPulseConfig

Para efetuar a interceptação de comandos para um dado ponto, primeiramente deve-se carregar no parâmetro dwVariableAddr o endereço da variável correspondente ao ponto que busca-se se interceptar comandos e então executar um pulso no parâmetro bExec. Uma vez recebido o comando, o function block informa que um comando foi interceptado através do parâmetro bCommandAvailable. As informações do comando interceptado são então preenchidas nos parâmetros de saída sCommand e eStatus conforme o tipo de comando que foi recebido. Esta operação depende somente do tipo de comando recebido, não importando o tipo de dado da variável para qual está se interceptando o comando. A interceptação é finalizada e então o bloco de função liberado para interceptar um novo comando quando sinalizado true no parâmetro bDone. Ainda deve ser indicado o resultado do processamento do comando em eCommandResult.

MODBUS – Tipos de Dados

A Tabela 4-41 mostra o tipo de variável suportada pela HX3040 para cada um dos tipos de dados do protocolo, aplicável para todos os dispositivos MODBUS.

Tipo de Relação	Tipos de variáveis aceitáveis
Coil	BOOL
Input Status	BOOL
Holding Registers	INT, UINT, WORD, REAL, DINT, UDINT, DWORD, LREAL, LINT, ULINT ou LWORD
Input Registers	INT, UINT, WORD, REAL, DINT, UDINT, DWORD, LREAL, LINT, ULINT ou LWORD

Tabela 4-41. Declaração de Variáveis para MODBUS

Nota:

Para escrita mascarada de Holding Register somente os tipos INT, UINT ou WORD podem ser usados. As variáveis do tipo REAL, DINT, UDINT e WORD possuem tamanho 2. Já as variáveis do tipo LREAL, LINT, ULINT e LWORD possuem tamanho 4. As demais variáveis possuem tamanho 1. No caso de mapeamentos com arrays, a quantidade de elementos dos mesmos será multiplicado pelo tamanho do tipo de dado informado anteriormente.

MODBUS RTU Mestre

Este protocolo está disponível para a UCP da Série Hadron Xtorm nos seus canais seriais. Ao selecionar esta opção no MasterTool Xtorm, a UCP passa a ser mestre da comunicação MODBUS, possibilitando o acesso a outros dispositivos com o mesmo protocolo, quando esta estiver em modo de execução (Modo Run).

Para configurar este protocolo, é necessário executar os seguintes passos:

1. Adicionar a instância do protocolo MODBUS RTU Mestre ao canal serial COM 1 ou COM 2 (ou ambos, em caso de duas redes de comunicação)
2. Para realizar esse procedimento, consultar o capítulo Configuração das Interfaces Seriais
3. Configurar a interface serial, escolhendo a velocidade de comunicação, o comportamento dos sinais RTS/CTS, a paridade, bits de parada do canal, entre outros através de um duplo clique sobre o canal serial COM 1 ou COM 2
4. Configurar os parâmetros gerais do protocolo MODBUS Mestre, como: tempos de atraso de envio e interframe mínimo
5. Adicionar e configurar dispositivos, definindo endereço do escravo, time-out de comunicação e número de retentativas de comunicação
6. Adicionar e configurar os mapeamentos MODBUS, especificando o nome da variável, tipo de dados, endereço inicial do dado, tamanho do dado e variável que receberá os dados de qualidade
7. Adicionar e configurar as requisições MODBUS, especificando a função desejada, o tempo de varredura da requisição, o endereço inicial (leitura/escrita), o tamanho dos dados (Leitura/Escrita), a variável que receberá os dados de qualidade, e a variável responsável por desabilitar a requisição

As descrições de cada configuração estão relacionadas a seguir, neste capítulo.

Parâmetros Gerais do Protocolo MODBUS Mestre

Os parâmetros gerais, encontrados na tela inicial de configuração do protocolo MODBUS (Figura 4-11), são definidos como:

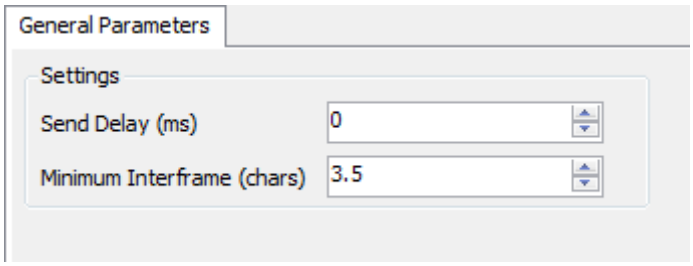


Figura 4-11. Tela de Configuração Parâmetros Gerais MODBUS RTU Mestre

Configuração	Descrição	Padrão de Fábrica	Possibilidades
Send Delay (ms)	Tempo de atraso para envio da resposta.	0	0 a 65535
Minimum Interframe (chars)	Tempo mínimo de silêncio entre diferentes frames.	3.5	3.5 a 100.0

Tabela 4-42. Configurações Gerais MODBUS RTU Mestre

Notas:

Send Delay: A resposta a uma requisição MODBUS pode causar problemas em certos momentos, como, por exemplo, na interface RS-485 ou outra half-duplex. Às vezes existe um atraso entre o tempo de resposta do escravo e o silêncio na linha física (atraso no escravo para colocar RTS em zero e colocar o transmissor RS-485 em alta impedância). Para resolver o problema, o mestre pode esperar o tempo determinado nesse campo antes de enviar a nova requisição. Caso contrário, os primeiros bytes transmitidos pelo mestre, durante a requisição, podem ser perdidos.

Minimum Interframe: A norma MODBUS define esse tempo como 3,5 caracteres, porém esse parâmetro é configurável para atender aos dispositivos que não estão de acordo com o padrão.

ATENÇÃO:
É recomendável que o usuário faça uso das duas estruturas de diagnósticos disponíveis para maior confiabilidade e abrangência dos diagnósticos, caso esteja consistindo a comunicação do Mestre RTU, através dos diagnósticos e comandos armazenados em variáveis do tipo T_DIAG_MODBUS_RTU_MASTER_1 as quais estão descritas na Tabela 4-43 e a estrutura de diagnósticos das requisições do tipo T_DIAG_MODBUS_RTU_MAPPING_1, as quais estão descritas na Tabela 4-49.

Variável de diagnóstico do tipo T_DIAG_MODBUS_RTU_MASTER_1.*		Tamanho	Descrição
Bits de diagnóstico:			
tDiag.*	bRunning	BOOL	O mestre está em execução.
	bNotRunning	BOOL	O mestre não está em execução (ver bit: bInterruptedByCommand).
	bInterruptedByCommand	BOOL	O bit bNotRunning foi habilitado, pois o mestre foi interrompido pelo usuário através de bits de comando.
	bConfigFailure	BOOL	Diagnóstico descontinuado.
	bRXFailure	BOOL	Diagnóstico descontinuado.

	bTXFailure	BOOL	Diagnóstico descontinuado.
	bModuleFailure	BOOL	Diagnóstico descontinuado.
	bDiag_7_reserved	BOOL	Reservado
Códigos de erro:			
eErrorCode		SERIAL_STATUS (BYTE)	0: não existem erros. 1: porta serial inválida. 2 :modo da porta serial inválido 3: baud rate inválido. 4: data bits inválido. 5: paridade inválida. 6: stop bits inválido. 7: parâmetro de sinal de modem inválido. 8: parâmetro de Threshold de RX da UART inválido. 9: parâmetro de time-out inválido. 10: porta serial ocupada. 11: erro de hardware na UART. 12: erro de hardware remoto. 20: tamanho do buffer de transmissão inválido. 21: método de sinal de modem inválido. 22: time-out de CTS = verdadeiro. 23: time-out de CTS = falso. 24: erro de time-out na transmissão. 30: tamanho do buffer de recepção inválido. 31: erro de time-out na recepção. 32: controle de fluxo configurado diferente de manual. 33: controle de fluxo inválido para a porta serial configurada. 34: recepção de dados não permitida no modo normal. 35: recepção de dados não permitida no modo estendido. 36: interrupção DCD não permitida. 37: interrupção CTS não permitida. 38: interrupção DSR não permitida. 39: porta serial não configurada. 50: erro interno na porta serial
Bits de comando, reiniciados automaticamente:			
tCommand.*	bStop	BOOL	Parar o mestre.
	bRestart	BOOL	Reiniciar o mestre.
	bResetCounter	BOOL	Reiniciar as estatísticas dos diagnósticos (contadores).
	bDiag_19_reserved	BOOL	Reservado
	bDiag_20_reserved	BOOL	Reservado
	bDiag_21_reserved	BOOL	Reservado
	bDiag_22_reserved	BOOL	Reservado
	bDiag_23_reserved	BOOL	Reservado
byDiag_03_reserved		BYTE	Reservado
Estatísticas de comunicação:			
tStat.*	wTXRequsts	WORD	Contador de requisições transmitidas pelo mestre (0 a 65535).
	wRXNormalResponses	WORD	Contador de respostas normais recebidas pelo mestre (0 a 65535).
	wRXExceptionResponses	WORD	Contador de respostas com códigos de exceção recebidas pelo mestre (0 a 65535).
	wRXIllegalResponses	WORD	Contador de respostas ilegais recebidas pelo mestre – sintaxe inválida, número insuficiente de bytes recebidos, CRC inválido – (0 a 65535).
	wRXOverrunErrors	WORD	Contador de erros de overrun durante a recepção – UART FIFO ou fila RX – (0 a 65535).
	wRXIncompleteFrames	WORD	Contador de respostas com erro de construção, paridade ou falha durante a recepção (0 a 65535).
	wCTSTimeOutErrors	WORD	Contador de erro de time-out no CTS, utilizando o handshake RTS/CTS, durante a transmissão (0 a 65535).
	wDiag_18_reserved	WORD	Reservado

Tabela 4-43. Diagnósticos MODBUS RTU Mestre

Notas:

Contadores: Todos os contadores dos diagnósticos do MODBUS RTU Mestre retornam à zero quando o valor limite 65535 é ultrapassado.

Configuração dos Dispositivos

A configuração dos dispositivos escravos, visualizados na Figura 4-12, seguem os seguintes parâmetros:

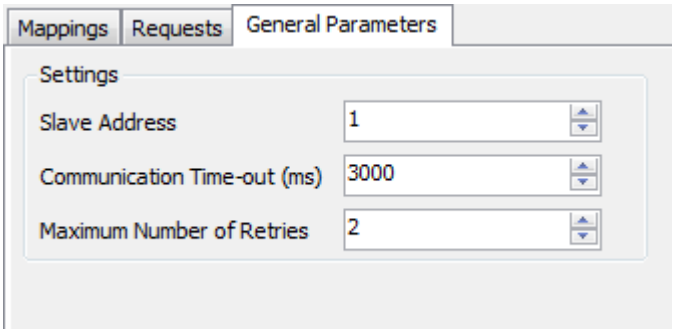


Figura 4-12. Tela de Configurações dos Parâmetros Gerais do Dispositivo

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Slave Address	Endereço do escravo MODBUS	1	0 a 255
Communication Time-out (ms)	Define o time-out do nível de aplicação	3000	10 a 65535
Maximum Number of Retries	Define o número de tentativas antes de reportar um erro de comunicação	2	0 a 9

Tabela 4-44. Configurações do Dispositivo

Notas:

Slave Address: De acordo com a Norma MODBUS, a faixa de endereços válidos para escravos é de 0 a 247, sendo os endereços 248 a 255 reservados. Quando o mestre envia um comando de escrita com o endereço configurado como zero, ele está realizando requisições broadcast na rede.

Communication Time-out: O time-out da comunicação é o tempo que o mestre aguardará por uma resposta do escravo à requisição. Para um dispositivo MODBUS RTU Mestre devem ser levadas em consideração ao menos as seguintes variáveis do sistema: o tempo que o escravo leva para transmitir o frame (de acordo com a taxa de transmissão), o tempo que o escravo leva para processar a requisição e o delay de envio da resposta caso seja configurado no escravo, o tempo de transmissão da resposta, possíveis atrasos causados por dispositivos intermediários, como repetidores. É recomendado que o time-out seja igual ou maior que o tempo para transmitir o frame somado ao delay de envio da resposta e a duas vezes o tempo de processamento da requisição.

A seguinte fórmula deve ser adotada para cálculo estimado de Tempo de Transmissão (ms):

Tempo de Transm(ms). = [1000 x (contagem de caracteres) x (bits por caracter)] / Taxa de Transm.

Maximum Number of Retries: Define o número de retentativas antes de reportar um erro de comunicação. Por exemplo, se o escravo não responder a um pedido e o mestre estiver configurado

para enviar três retentativas, o número do contador de erros será incrementado por 1 unidade ao final da execução destas três retentativas.

Configuração dos Mapeamentos

A configuração das relações MODBUS, visualizadas na Figura 4-13, segue os parâmetros descritos na Tabela 4-45:

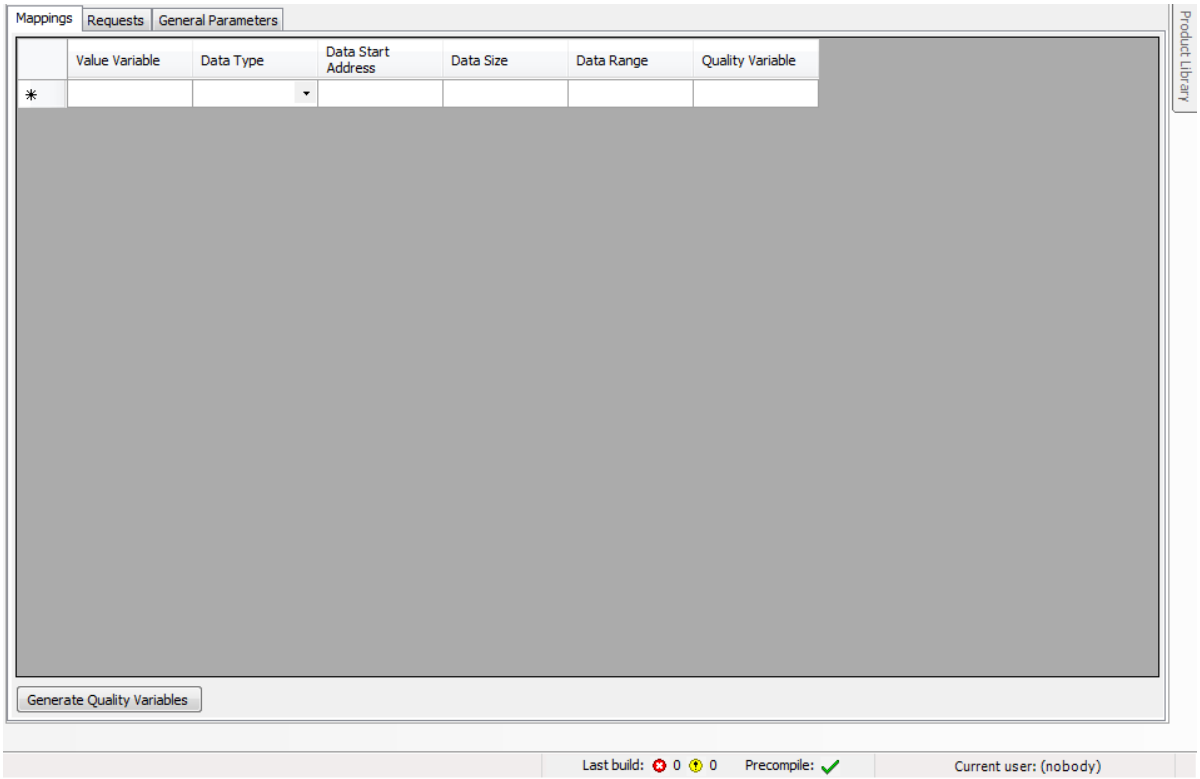


Figura 4-13. Tela de Mapeamentos de Dados MODBUS

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Value Variable	Nome da variável simbólica	-	Nome de uma variável declarada em um programa ou GVL
Data Type	Tipo de dado MODBUS	-	Coil Escrita(1 bit) Coil Leitura (1 bit) Holding Register Escrita (16 bits) Holding Register Leitura (16 bits) Holding Register – Mascara AND (16 bits) Holding Register – Mascara OR (16 bits) Input Register (16 bits) Input Status (1 bit)
Data Start Address	Endereço inicial dos dados MODBUS	-	1 a 65536
Data Size	Tamanho do dado MODBUS	-	1 a 65536
Data Range	Faixa de endereços do dado configurado	-	-
Quality variable	Nome da variável que vai conter os dados de qualidade do mapeamento MODBUS	-	Nome de uma variável declarada em um programa ou GVL. A variável deve ser do tipo DWORD
Generate Quality Variables	Gera automaticamente na GVL IOQualities variáveis do tipo QUALITY para cada Variável de Valor declarada. Para mais informações acesse o capítulo Conversões de Qualidade.	-	-

Tabela 4-45. Configuração dos Mapeamentos MODBUS

Notas:

riting

Tipo de Dado	Tamanho [bits]	Descrição
Coil Escrita	1	Saída digital de escrita.
Coil Leitura	1	Saída digital de leitura.
Holding Register de Escrita	16	Saída analógica de escrita.
Holding Register de Leitura	16	Saída analógica de leitura.
Holding Register com mascara AND	16	Saída analógica que pode ser lida ou escrita com máscara AND.
Holding Register com mascara OR	16	Saída analógica que pode ser lida ou escrita com máscara OR.
Input Register	16	Entrada analógica que pode ser apenas lida.
Input Status	1	Entrada digital que pode ser apenas lida.

Tabela 4-46. Tipos de Dados Suportados no MODBUS RTU Mestre

Data Start Address: Endereço inicial do dado de um mapeamento MODBUS.

Data Size: O valor de Tamanho especifica a quantidade máxima de dados que uma relação MODBUS poderá acessar, a partir do endereço inicial. Sendo assim, para ler uma faixa de endereços contínua, é necessário que todos os endereços estejam declarados em uma única relação. Este campo varia de acordo com o tipo de dado MODBUS configurado.

Data Range: Este campo mostra ao usuário a faixa de endereços de memória utilizada pela relação MODBUS.

Quality Variable: Este campo é utilizado para especificar uma variável que vai conter os dados da qualidade da relação MODBUS.

ATENÇÂO:
Caso o usuário utilize uma variável que abranja o endereço 5 a 10 de uma função declarada na aba de Mapeamentos, será válida a leitura/escrita de Requisições que atinjam as extremidades ou parte do Mapeamento.
Exemplo, a leitura/escrita dessa função do endereço 1 ao 5, onde somente o endereço 5 está declarado, este endereço receberá valores válidos e o restante será desconsiderado, sem valores válidos.

Configuração das Requisições

A configuração das requisições MODBUS, visualizadas na Figura 4-14 , segue os parâmetros descritos na Tabela 4-47:

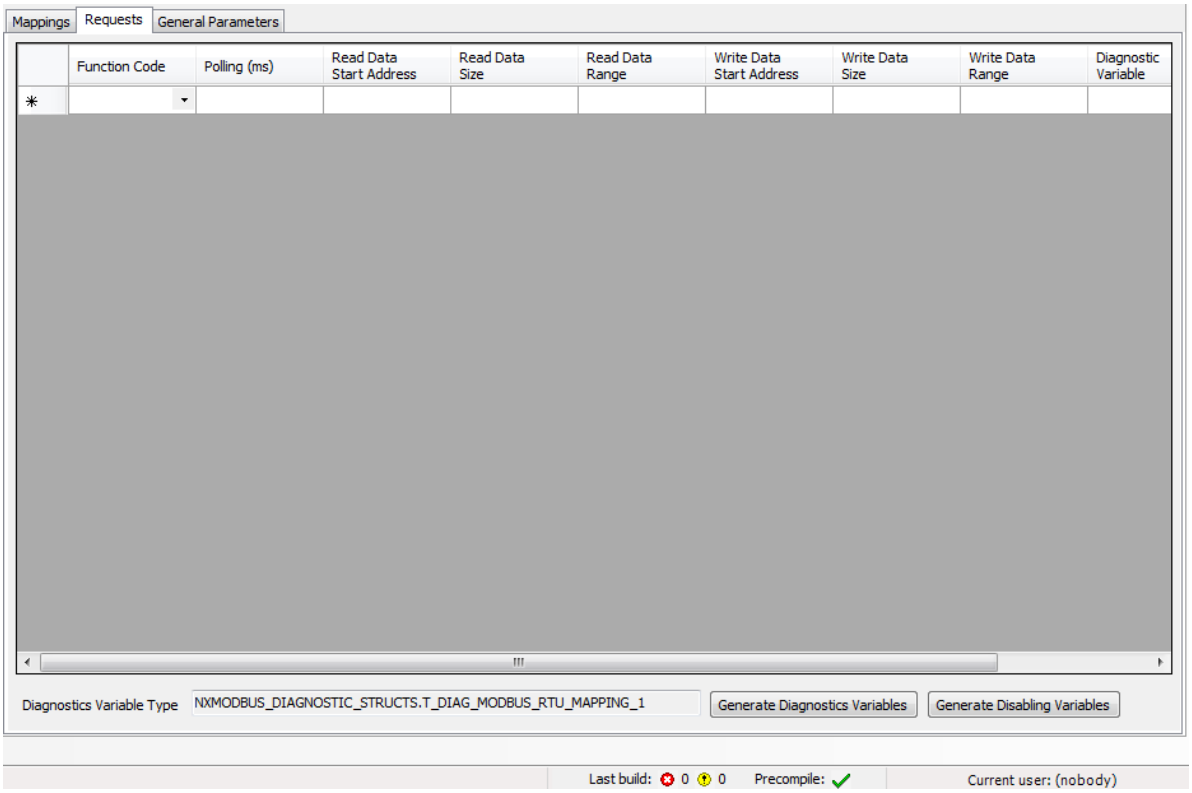


Figura 4-14. Tela de Requisições de Dados MODBUS Mestre

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Function Code	Tipo de função MODBUS	-	FC01 – Leitura de Coils FC02 – Leitura de Input Status FC03 – Leitura de Holding Registers FC04 – Leitura de Input Registers FC05 – Escrita de um Coil FC06 – Escrita de um Holding Register FC15 – Escrita de múltiplos Coils FC16 – Escrita de Holding Registers FC22 – Máscara de Escrita do Register FC23 – Leitura/Escrita de Holding Register
Polling (ms)	Período de comunicação (ms)	-	0 a 3600000
Read Data Start Address	Endereço inicial dos dados de leitura MODBUS	-	1 a 65536
Read Data Size	Tamanho dos dados de leitura MODBUS	-	Depende da função utilizada
Read Data Range	Faixa de endereço dos dados de leitura MODBUS	-	0 a 2147483646
Write Data Start Address	Endereço inicial dos dados de escrita MODBUS	1	1 a 65536
Write Data Size	Tamanho dos dados de escrita MODBUS	-	Depende da função utilizada
Write Data Range	Faixa de endereço dos dados de escrita MODBUS	-	0 a 2147483647
Diagnostic Variable	Nome da variável de diagnóstico	-	Nome de uma variável declarada em um programa ou GVL
Disabling Variable	Variável utilizada para desabilitar a relação MODBUS	-	Campo destinado a variável simbólica utilizada para desabilitar, individualmente, as requisições MODBUS configuradas. Esta variável deve ser do tipo BOOL. A variável pode ser simples ou elemento de array e pode estar em estruturas.

Tabela 4-47. Configuração das Relações MODBUS Mestre

Notas:

Generate Diagnostic Variables: As variáveis simbólicas de diagnósticos podem ser geradas automaticamente através do botão "Gerar Variáveis de Diagnósticos". Ao clicar nesse botão, será criada uma variável do tipo" T_DIAG_MODBUS_RTU_MAPPING_1", na GVL "ReqDiagnostics", para cada uma das requisições.

Generate Disabling Variables: As variáveis simbólicas de desabilitação podem ser geradas automaticamente através do botão "Gerar Variáveis de Desabilitação". Ao clicar nesse botão, será criada uma variável do tipo BOOL, na GVL "Disables", para cada uma das requisições.

Configuração das relações MODBUS: O número de configurações, padrão de fábrica e os valores da coluna possibilidades, podem variar de acordo com o tipo de dado e função MODBUS (FC).

Function Code: As funções MODBUS (FC) disponíveis são os seguintes:

Tipo de Função	Código		Descrição
	DEC	HEX	
Acesso a Variáveis	1	0x01	Leitura de Coils (FC 01)
	2	0x02	Leitura de input status (FC 02)
	3	0x03	Leitura de holding registers (FC 03)
	4	0x04	Leitura de input registers (FC 04)
	5	0x05	Escrita de um Coil (FC 05)
	6	0x06	Escrita de um holding register (FC 06)
	15	0x0F	Escrita de múltiplos Coils (FC 15)
	16	0x10	Escrita de holding registers (FC 16)
	22	0x16	Máscara de escrita do register (FC 22)
	23	0x17	Leitura/escrita holding registers (FC 23)

Tabela 4-48. Funções MODBUS Suportadas pela UCP Hadron Xtorm

Polling: Este parâmetro indica com que frequência à comunicação definida por esta requisição deve ser executada. Ao ser finalizada uma comunicação será aguardado um tempo igual ao configurado no campo varredura e, após, será executada uma nova comunicação.

Read Data Start Address: Campo destinado ao endereço inicial dos dados de leitura MODBUS.

Read Data Size: O valor mínimo para o tamanho dos dados de leitura é 1 e o valor máximo depende da função MODBUS (FC) utilizada, conforme abaixo:

1. Leitura de Coils (FC 1): 2000
2. Leitura de Input Status (FC 2): 2000
3. Leitura de Holding Registers (FC 3): 125
4. Leitura de Input Registers (FC 4): 125
5. Leitura/Escrita de Holding Registers (FC 23): 121

Read Data Range: Este campo mostra a faixa de dados de leitura MODBUS configurada para cada requisição. O endereço inicial de leitura, somado ao tamanho do dado de leitura resultará na faixa de dados de leitura de cada uma das requisições.

Write Data Start Address: Campo destinado ao endereço inicial dos dados de escrita MODBUS.

Write Data Size: O valor mínimo para o tamanho dos dados de escrita é 1 e o valor máximo depende da função MODBUS (FC) utilizada, conforme abaixo:

1. Escrita de Um Coil (FC 5): 1
2. Escrita de Um Holding Registers (FC 6): 1
3. Escrita de Múltiplos Coils (FC 15): 1968
4. Escrita de Holding Registers (FC 16): 123
5. Máscara de Escrita do Register (FC 22): 1

6. Leitura/Escrita de Holding Registers (FC 23): 121

Write Data Range: Este campo mostra a faixa de dados de escrita MODBUS configurada para cada requisição. O endereço inicial de escrita, somado ao tamanho do dado de escrita resultará na faixa de dados de escrita de cada uma das requisições.

Generate Diagnostic Variables: Os diagnósticos da requisição MODBUS configurada, são armazenados em variáveis do tipo T_DIAG_MODBUS_RTU_MAPPING_1. a qual está descrita na Tabela 4-49.

Variável de diagnóstico do tipo T_DIAG_MODBUS_RTU_MAPPING_1.*		**Tamanho**	**Descrição**
Bits de estado da comunicação:			
byStatus.*	bCommIdle	BOOL	Comunicação inativa (aguardando ser constada).
	bCommExecuting	BOOL	Comunicação ativa.
	bCommPostponed	BOOL	Comunicação adiada, pois o número máximo de requisições simultâneas foi atingido. As comunicações adiadas serão executadas na mesma sequência em que elas foram requisitadas, evitando a indeterminação. O tempo gasto nesse estado não é contabilizado para efeitos de time-out. Os bits bCommIdle e bCommExecuting são falsos quando o bit bCommPostponed é verdadeiro.
	bCommDisabled	BOOL	Comunicação desabilitada. O bit bCommIdle é reiniciado nessa condição.
	bCommOk	BOOL	Comunicação finalizada anteriormente foi realizada com sucesso.
	bCommError	BOOL	Comunicação finalizada anteriormente teve um erro. Verificar código de erro.
	bDiag_6_reserved	BOOL	Reservado
	bDiag_7_reserved	BOOL	Reservado
Último código de erro (habilitado quando o bCommError = verdadeiro):			
eLastErrorCode		MASTER_ERROR_CODE (BYTE)	ERR_EXCEPTION (1): resposta reportada em um código de exceção (ver eLastExceptionCode = Código de Exceção). ERR_CRC (2): resposta com CRC inválido. ERR_ADDRESS (3): endereço MODBUS não encontrado. O endereço que respondeu à requisição foi diferente do esperado. ERR_FUNCTION (4): código inválido da função. A função recebida na resposta foi diferente da esperada pela requisição. ERR_FRAME_DATA_COUNT (5): a quantidade de dados da resposta foi diferente do esperado. ERR_NOT_ECHO (7): resposta não é eco da pergunta (FC 5 e 6). ERR_REFERENCE_NUMBER (8): número de referência inválido (FC 15 e 16). ERR_INVALID_FRAME_SIZE (9): resposta menor do que a esperada. ERR_SEND (21): erro durante a fase de transmissão. ERR_RECEIVE (22): erro durante a fase de recepção. ERR_SEND_TIMEOUT (41): time-out no nível de aplicação durante a transmissão. ERR_RECEIVE_TIMEOUT (42): time-out no nível de aplicação enquanto aguarda resposta. ERR_CTS_OFF_TIMEOUT (43): time-out

		enquanto aguarda CTS = falso na transmissão. ERR_CTS_ON_TIMEOUT (44): time-out enquanto aguarda CTS = verdadeiro na transmissão. NO_ERROR (128): sem erro desde a inicialização.
Último código de exceção recebido pelo mestre:		
eLastExceptionCode	MODBUS_EXCEPTION (BYTE)	NO_EXCEPTION (0): não ocorreu exceção. FUNCTION_NOT_SUPPORTED (1): função MODBUS não suportada pelo equipamento. MAPPING_NOT_FOUND (2): mapeamento não encontrado. ILLEGAL_VALUE (3): valor ilegal recebido. ACCESS_DENIED (128)*: acesso não permitido ao equipamento. MAPPING_DISABLED (129)*: mapeamento desabilitado pelo usuário ou lógica de controle IGNORE_FRAME (255)*: frame invalido.
Estatísticas de comunicação:		
byDiag_3_reserved	BYTE	Reservado.
wCommCounter	WORD	Contador de comunicações finalizadas, com ou sem erros. O usuário pode testar quando a comunicação foi finalizada testando a variação desse contador. Quando o valor 65535 é atingido, o contador retorna à zero.
wCommErrorCounter	WORD	Contador de comunicações finalizadas com erros. Quando o valor 65535 é atingido, o contador retorna a zero.

Tabela 4-49. Diagnósticos das Relações MODBUS

Códigos de exceção (eLastExceptionCode): As definições dos códigos de exceção 128, 129 e 255 são válidas apenas na utilização de escravos Altus. Para escravos de outros fabricantes esses códigos de exceção podem ter significados diferentes.

Generate Disabling Variables: Campo destinado a variável utilizada para desabilitar, individualmente, as requisições MODBUS configuradas no espaço Requisições. A requisição é desabilitada quando a variável, correspondente a requisição, for igual a 1, caso contrário, o a requisição está habilitada.

ATENÇÃO:
Diferentemente de outras tarefas de uma aplicação, quando for atingida uma marca de depuração na MainTask, a tarefa de uma instância MODBUS RTU Mestre, e qualquer outra tarefa MODBUS, irão parar de ser executada no momento em que tentar efetuar uma escrita em uma área de memória. Isto ocorre para manter a consistência dos dados das áreas de memória enquanto a MainTask não estiver em execução.

MODBUS RTU Escravo

Este protocolo está disponível para a UCP da Série Hadron Xtorm nos seus canais seriais. Ao selecionar esta opção no MasterTool Xtorm, a UCP passa a ser escravo da comunicação MODBUS, permitindo a conexão com dispositivos mestre MODBUS RTU. Este protocolo somente está disponível quando a UCP estiver em modo de execução (Modo Run).

Para configurar este protocolo, é necessário executar os seguintes passos:

1. Adicionar a instância do protocolo MODBUS RTU Escravo ao canal serial COM 1 ou COM 2 (ou ambos, em caso de duas redes de comunicação)
2. Para realizar esse procedimento, consultar o capítulo Inserindo uma Instância de Protocolo

3. Configurar a interface serial, escolhendo a velocidade de comunicação, o comportamento dos sinais RTS/CTS, a paridade, os bits de parada do canal, entre outros através de um duplo clique sobre o canal serial COM 1 ou COM 2
4. Configurar os parâmetros gerais do protocolo MODBUS escravo, como: Endereço do Escravo e tempos de comunicação (disponível no botão de configurações avançadas do Escravo)
5. Adicionar e configurar as relações MODBUS, especificando o nome da variável, tipo de dado MODBUS, o endereço inicial do dado e o tamanho do dado

As descrições de cada configuração estão relacionadas a seguir, neste capítulo.

Parâmetros Gerais do Protocolo MODBUS Slave

Os parâmetros gerais, encontrados na tela inicial de configuração do protocolo MODBUS (Figura 4-15), são definidos como:

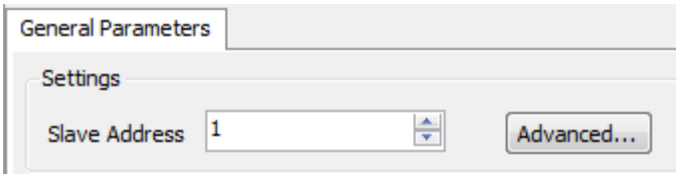


Figura 4-15. Tela de Configuração do Escravo

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Slave Address	Endereço do escravo MODBUS	1	1 a 255

Tabela 4-50. Configurações do Escravo

- Tempos de comunicação:

Os tempos de comunicação do protocolo MODBUS Escravo, encontrados no botão “Advanced...” da tela de configuração, estão divididos em: Ciclo da Tarefa, Atraso do Envio e Interframe Mínimo.

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Task Cycle (ms)	Tempo para execução da instância dentro do ciclo, sem considerar o tempo de execução da mesma	50	20 a 100
Send Delay (ms)	Tempo de atraso para envio da resposta	0	0 a 65535
Minimum Interframe (chars)	Tempo mínimo de silêncio entre diferentes frames	3.5	3.5 a 100.0

Tabela 4-51. Configurações Avançadas MODBUS Escravo

Notas:

Task Cycle: O usuário deverá ter cuidado a alterar esse parâmetro, pois o mesmo interfere diretamente no tempo de resposta, volume de dados por varredura e, principalmente, no balanceamento dos recursos da UCP entre comunicações e outras tarefas.

Send Delay: A resposta à uma requisição MODBUS pode causar problemas em certos momentos, como, por exemplo, na interface RS-485 ou outra half-duplex. Às vezes existe um atraso entre o tempo da requisição do mestre e o silêncio na linha física (atraso no mestre para colocar RTS em zero e colocar o transmissor RS-485 em alta impedância). Para resolver o problema, o escravo pode esperar o tempo determinado nesse campo antes de enviar a resposta. Caso contrário, os primeiros bytes transmitidos pelo escravo, durante a resposta, podem ser perdidos.

Minimum Interframe: A norma MODBUS define esse tempo como 3,5 caracteres, porém esse parâmetro é configurável para atender aos dispositivos que não estão de acordo com o padrão.

Os diagnósticos e comandos do protocolo MODBUS RTU Slave configurado, são armazenados em variáveis do tipo T_DIAG_MODBUS_RTU_SLAVE_1 as quais estão descritas na Tabela 4-52:

Variável de diagnósticos do tipo T_DIAG_MODBUS_RTU_SLAVE_1 *.		Tamanho	Descrição
Bits de diagnóstico:			
tDiag.*	bRunning	BOOL	O escravo está em execução.
	bNotRunning	BOOL	O escravo não está em execução (ver bit: bInterruptedByCommand).
	bInterruptedByCommand	BOOL	O bit bNotRunning foi habilitado, pois o escravo foi interrompido pelo usuário através de bits de comando.
	bConfigFailure	BOOL	Diagnóstico descontinuado.
	bRXFailure	BOOL	Diagnóstico descontinuado.
	bTXFailure	BOOL	Diagnóstico descontinuado.
	bModuleFailure	BOOL	Diagnóstico descontinuado.
	bDiag_7_reserved	BOOL	Reservado
Códigos de erro:			
eErrorCode		SERIAL_STATUS (BYTE)	0: não existem erros. 1: porta serial inválida. 2 :modo da porta serial inválido 3: baud rate inválido. 4: data bits inválido. 5: paridade inválida. 6: stop bits inválido. 7: parâmetro de sinal de modem inválido. 8: parâmetro de Threshold de RX da UART inválido. 9: parâmetro de time-out inválido. 10: porta serial ocupada. 11: erro de hardware na UART. 12: erro de hardware remoto. 20: tamanho do buffer de transmissão inválido. 21: método de sinal de modem inválido. 22: time-out de CTS = verdadeiro. 23: time-out de CTS = falso. 24: erro de time-out na transmissão. 30: tamanho do buffer de recepção inválido. 31: erro de time-out na recepção. 32: controle de fluxo configurado diferente de manual. 33: controle de fluxo inválido para a porta serial configurada. 34: recepção de dados não permitida no modo normal. 35: recepção de dados não permitida no modo estendido. 36: interrupção DCD não permitida. 37: interrupção CTS não permitida. 38: interrupção DSR não permitida. 39: porta serial não configurada. 50: erro interno na porta serial
Bits de comando, reiniciados automaticamente:			
tCommand.*	bStop	BOOL	Parar o escravo.
	bRestart	BOOL	Reiniciar o escravo.
	bResetCounter	BOOL	Reiniciar as estatísticas dos diagnósticos (contadores).

	bDiag_19_reserved	BOOL	Reservado
	bDiag_20_reserved	BOOL	Reservado
	bDiag_21_reserved	BOOL	Reservado
	bDiag_22_reserved	BOOL	Reservado
	bDiag_23_reserved	BOOL	Reservado
byDiag_03_reserved		BYTE	Reservado
Estatísticas de comunicação:			
tStat.*	wRXRequsts	WORD	Contador de requisições normais recebidas pelo escravo e respondidas normalmente. No caso de um comando broadcast, esse contador é incrementado, porém não é transmitida a resposta (0 a 65535).
	wTXExceptionResponses	WORD	Contador de requisições normais recebidas pelo escravo e respondidas com códigos de exceção. No caso de um comando broadcast, esse contador é incrementado, porém não é transmitida a resposta (0 a 65535). Códigos de exceção: 1: o código da função (FC) é legal, porém não suportado. 2: relação não encontrada nestes dados MODBUS. 3: valor ilegal para o endereço. 128: o mestre/cliente não tem direito de escrita ou leitura. 129: a relação MODBUS está desabilitada.
	wRXFrames	WORD	Contador de frames recebidos pelo escravo. Considera-se frame, algo que é processado e seguido de um período mínimo de silêncio interframes (Interframe Mínimo), ou seja, uma mensagem ilegal também é computada (0 a 65535).
	wRXIllegalRequests	WORD	Contador de requisições ilegais. Estes são frames que iniciam com o endereço 0 (broadcast) ou com o endereço MODBUS do escravo, mas não são requisições legais – sintaxe inválida, frames menores, CRC inválido – (0 a 65535).
	wRXOverrunErrors	WORD	Contador de frames com erros de overrun durante a recepção – UART FIFO ou fila RX – (0 a 65535).
	wRXIncompleteFrames	WORD	Contador de frames com erro de construção, paridade ou falha durante a recepção (0 a 65535).
	wCTSTimeOutErrors	WORD	Contador de erro de time-out no CTS, utilizando o handshake RTS/CTS, durante a transmissão (0 a 65535).
	wDiag_18_reserved	WORD	Reservado.

Tabela 4-52. Diagnósticos MODBUS RTU Slave

Nota:

Contadores: Todos os contadores dos diagnósticos do MODBUS RTU Escravo retornam à zero quando o valor limite 65535 é ultrapassado.

Configuração dos Mapeamentos

A configuração das relações MODBUS, visualizadas na Figura 4-16, segue os parâmetros descritos na Tabela 4-53:

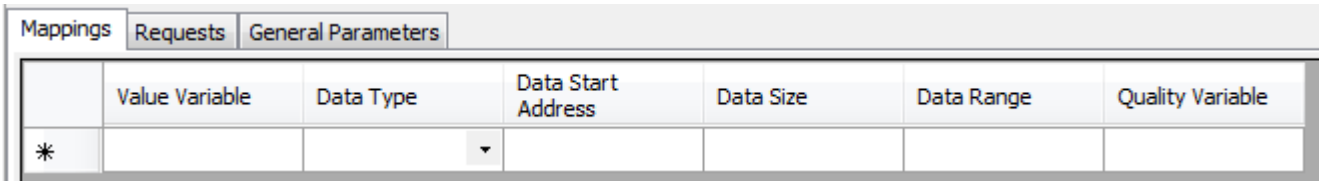


Figura 4-16. Tela de Mapeamentos de Dados MODBUS

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Value Variable	Nome da variável simbólica	-	Nome de uma variável declarada em um programa ou GVL
Data Type	Tipo de dado MODBUS	-	Coil (1 bit) Input Status (1 bit) Holding Register (16 bits) Input Register (16 bits)
Data Start Address	Endereço inicial dos dados MODBUS	-	1 a 65536
Data Size	Tamanho do dado MODBUS	-	1 a 65536
Data Range	Faixa de endereços do dado configurado	-	-
Quality Variable	Nome da variável simbólica que armazena a qualidade do mapeamento, que deve ser do tipo QUALITY, disponível na biblioteca LibDataTypes		A variável pode ser simples, array ou elemento de array e pode estar em estruturas

Tabela 4-53. Configuração dos Mapeamentos MODBUS

Notas:

Value Variable: Esse campo é utilizado para especificar uma variável simbólica na relação MODBUS.

Data Type: Esse campo é utilizado para especificar o tipo de dado utilizado na relação MODBUS.

Tipo de Dado	Tamanho [bits]	Descrição
Coil	1	Saída digital que pode ser lida ou escrita.
Input Status	1	Entrada digital que pode ser apenas lida.
Holding Register	16	Saída analógica que pode ser lida ou escrita.
Input Register	16	Entrada analógica que pode ser apenas lida.

Tabela 4-54. Tipos de Dados Suportados no MODBUS RTU Escravo

Data Start Address: Endereço inicial do dado de uma relação MODBUS.

Data Size: O valor de Tamanho especifica a quantidade máxima de dados que uma relação MODBUS poderá acessar, a partir do endereço inicial. Sendo assim, para ler uma faixa de endereços contínua, é necessário que todos os endereços estejam declarados em uma única relação. Este campo varia de acordo com o tipo de dado MODBUS configurado.

Data Range: Este campo mostra ao usuário a faixa de endereços de memória utilizada pela relação MODBUS.

Máximo Tamanho de Relação: Cada COM tem limite de 20000 pontos de comunicação, ou seja, caso o usuário utilize os 20000 pontos em uma única relação na COM1, não será possível utilizar mais pontos de comunicação nesta COM. Entretanto uma boa prática é utilizar quantidades e tipos de relações apropriadas às necessidades da aplicação. Para mais informações ver capítulo de Pontos Duplos.

.

ATENÇÃO:
Diferentemente de outras tarefas de uma aplicação, quando for atingida uma marca de depuração na MainTask, a tarefa de uma instância MODBUS RTU Escravo, e qualquer outra tarefa MODBUS, irão parar de ser executada no momento em que tentar efetuar uma escrita em uma área de memória. Isto ocorre para manter a consistência dos dados das áreas de memória enquanto a MainTask não estiver em execução.

MODBUS Ethernet

A rede de comunicação multimestre permite que a UCP Hadron Xtorm leia ou escreva variáveis MODBUS em outros controladores ou IHMs compatíveis com os protocolos MODBUS TCP ou MODBUS RTU via TCP. A UCP Hadron Xtorm pode, simultaneamente, ser cliente e servidor em uma mesma rede de comunicação, ou até mesmo ter mais instâncias associadas à interface Ethernet, indiferente se as mesmas são MODBUS TCP ou MODBUS RTU via TCP, conforme descreve a Tabela 4-23.

A Figura 4-17 representa algumas das possibilidades de comunicação utilizando-se o protocolo MODBUS TCP simultaneamente com o protocolo MODBUS RTU via TCP.

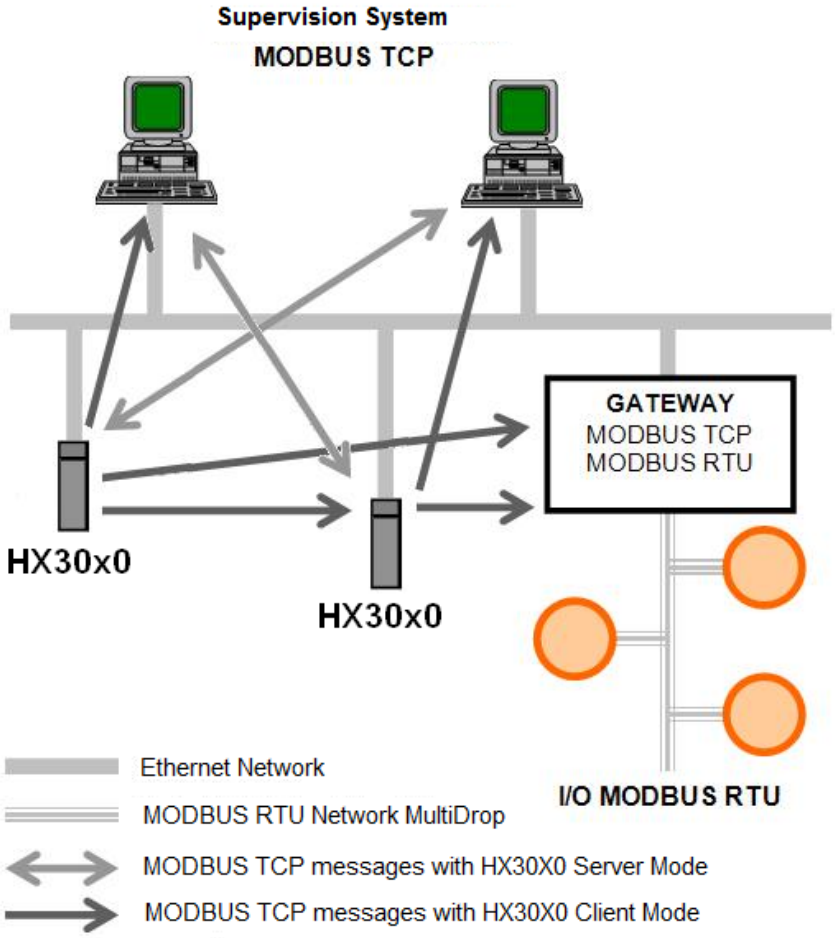


Figura 4-17. Rede de Comunicação MODBUS TCP

A associação de variáveis MODBUS com variáveis simbólicas da UCP é realizada pelo usuário através da definição de relações via configurador MasterTool Xtorm. Podem ser definidas até 32 relações para o modo servidor e até 128 relações para o modo cliente. Uma relação, em modo servidor, pode definir uma grande área de dados MODBUS e torná-la disponível para vários clientes. As relações em modo cliente, por outro lado, devem respeitar o tamanho máximo de dados de uma função MODBUS: 125 registradores (input registers ou holding registers) ou 2000 bits (coils ou input status). Essas informações são detalhadas na descrição de cada protocolo.

Todas as relações, em modo cliente ou servidor, podem ser desabilitadas através de variáveis simbólicas identificadas na coluna Desabilitação no MasterTool Xtorm. A desabilitação ocorre através destes bits específicos, afetando relações específicas.

Para as relações em modo servidor, podem ser definidos conjuntos de endereços IPs com permissão de escrita e leitura, chamados de filtros. Isto é feito através da definição de um endereço de rede IP e

de uma máscara de subrede, resultando em um grupo de IPs clientes que podem escrever e ler nas variáveis da relação. Essas informações são detalhadas na descrição do protocolo MODBUS Ethernet Servidor.

Quando o protocolo MODBUS TCP é utilizado no modo cliente, pode-se usufruir da característica de múltiplas requisições, utilizando a mesma conexão TCP para acelerar a comunicação com os servidores. Quando esta característica não for desejada ou não for suportada pelo servidor, ela pode ser desabilitada (ação em nível de relação). É importante destacar que o número máximo de conexões TCP entre cliente e servidor é 63, sendo que se alguns parâmetros forem alterados, comunicações inativas podem ser fechadas, possibilitando a abertura de novas conexões.

MODBUS Ethernet Cliente

Este protocolo está disponível para a UCP da Série Hadron Xtorm nos seus canais Ethernet. Ao selecionar esta opção no MasterTool Xtorm, a UCP passa a ser cliente da comunicação MODBUS, possibilitando o acesso a outros dispositivos com o mesmo protocolo, quando esta estiver em modo de execução (Modo Run).

Para configurar este protocolo, é necessário executar os seguintes passos:

- Adicionar a instância do protocolo MODBUS Ethernet Cliente a um dos canais Ethernet disponíveis (NET 1 ao NET 6)
- Para realizar esse procedimento, consultar o capítulo Inserindo uma Instância de Protocolo
- Configurar a interface Ethernet. Para realizar esse procedimento, consultar o capítulo Configuração das Interfaces Ethernet
- Configurar os parâmetros gerais do protocolo MODBUS Cliente, com o protocolo TCP ou RTU via TCP
- Adicionar e configurar dispositivos, definindo endereço IP, porta, endereço do escravo e time-out de comunicação (disponível no botão de configurações avançadas do Dispositivo)
- Adicionar e configurar os mapeamentos MODBUS, especificando o nome da variável, tipo de dados, endereço inicial do dado, tamanho do dado e variável que receberá os dados de qualidade
- Adicionar e configurar as requisições MODBUS, especificando a função desejada, o tempo de varredura da requisição, o endereço inicial (leitura/escrita), o tamanho dos dados (Leitura/Escrita), a variável que receberá os dados de qualidade, e a variável responsável por desabilitar a requisição

As descrições de cada configuração estão relacionadas a seguir, neste capítulo.

Parâmetros Gerais do Protocolo MODBUS Cliente

Os parâmetros gerais, encontrados na tela inicial de configuração do protocolo MODBUS (Figura 4-15), são definidos como:

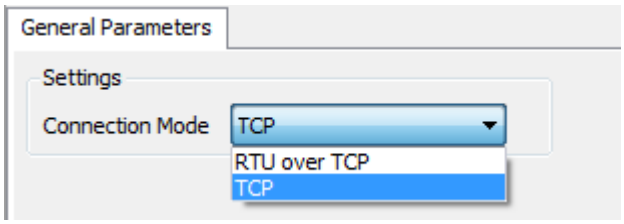


Figura 4-18. Tela de Configuração Parâmetros Gerais MODBUS TCP Cliente

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Connection Mode	Seleção do protocolo	TCP	RTU over TCP TCP

Tabela 4-55 Configurações Gerais MODBUS RTU Mestre

Os diagnósticos e comandos do protocolo MODBUS TCP Cliente configurado, são armazenados em variáveis do tipo T_DIAG_MODBUS_ETH_CLIENT_1 as quais estão descritas na Tabela 4-56:

Variável de diagnóstico do tipo T_DIAG_MODBUS_ETH_CLIENT_1.*		Tamanho	Descrição
Bits de diagnóstico:			
tDiag.*	bRunning	BOOL	O cliente está em execução.
	bNotRunning	BOOL	O cliente não está em execução (ver bit bInterruptedByCommand).
	bInterruptedByCommand	BOOL	O bit bNotRunning foi habilitado pois o cliente foi interrompido pelo usuário através de bits de comando.
	bConfigFailure	BOOL	Diagnóstico descontinuado.
	bRXFailure	BOOL	Diagnóstico descontinuado.
	bTXFailure	BOOL	Diagnóstico descontinuado.
	bModuleFailure	BOOL	Diagnóstico descontinuado.
	bDiag_7_reserved	BOOL	Reservado
byDiag_1_reserved		BYTE	Reservado
Bits de comando, reiniciados automaticamente:			
tCommand.*	bStop	BOOL	Parar o cliente.
	bRestart	BOOL	Reiniciar o cliente.
	bResetCounter	BOOL	Reiniciar as estatísticas dos diagnósticos (contadores).
	bDiag_19_reserved	BOOL	Reservado
	bDiag_20_reserved	BOOL	Reservado
	bDiag_21_reserved	BOOL	Reservado
	bDiag_22_reserved	BOOL	Reservado
	bDiag_23_reserved	BOOL	Reservado
byDiag_03_reserved		BYTE	Reservado
Estatísticas de comunicação:			
tStat.*	wTXRequsts	WORD	Contador de requisições transmitidas pelo cliente (0 a 65535).
	wRXNormalResponses	WORD	Contador de respostas normais recebidas pelo cliente (0 a 65535).
	wRXExceptionResponses	WORD	Contador de respostas com códigos de exceção (0 a 65535).
	wRXIllegalResponses	WORD	Contador de respostas ilegais recebidas pelo cliente – sintaxe inválida, CRC inválido ou número insuficiente de bytes recebidos (0 a 65535).
	wDiag_12_reserved	WORD	Reservado
	wDiag_14_reserved	WORD	Reservado
	wDiag_16_reserved	WORD	Reservado
	wDiag_18_reserved	WORD	Reservado

Tabela 4-56. Diagnósticos MODBUS TCP Cliente

Notas:

Contadores: Todos os contadores dos diagnósticos do MODBUS TCP Cliente retornam à zero quando o valor limite 65535 é ultrapassado.

Configuração dos Dispositivos

A configuração dos dispositivos escravos, visualizados na Tabela 4-57, seguem os seguintes parâmetros:

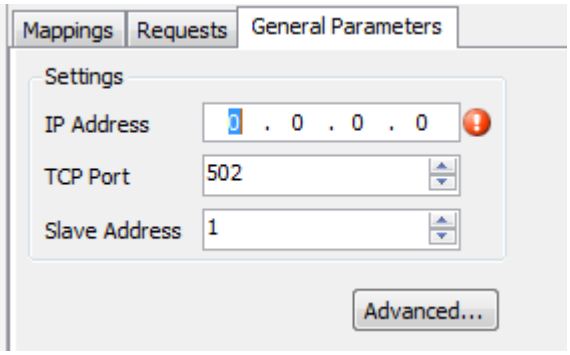


Figura 4-19. Tela de Configurações dos Parâmetros Gerais do Dispositivo

Configurações	Descrição	Padrão de Fábrica	Possibilidades
IP Address	Endereço IP do servidor	0.0.0.0	1.0.0.1 a 223.255.255.254
TCP Port	Porta TCP	502	2 a 65534
Slave Address	Endereço do Escravo MODBUS	1	0 a 255

Tabela 4-57. Configurações Gerais MODBUS RTU Cliente

Notas:

IP Address: Endereço IP do dispositivo servidor Modbus.

TCP Port: Caso sejam adicionadas várias instâncias do protocolo em uma única interface Ethernet, diferentes portas TCP devem ser selecionadas para cada instância. Algumas portas TCP, são reservadas, veja capítulo Portas TCP Reservadas.

Slave Address: De acordo com a Norma MODBUS, a faixa de endereços válidos para escravos é de 0 a 247, sendo os endereços 248 a 255 reservados. Quando o mestre envia um comando de escrita com o endereço configurado como zero, ele está realizando requisições broadcast na rede.

- Time-out de comunicação:

Os parâmetros de Time-out de comunicação do protocolo MODBUS Cliente, encontrados no botão "Avançado..." da tela de configuração, estão divididos em: número máximo de requisições simultâneas, Time-out de Comunicação, Modo de time-out de conexão e tempo de inatividade.

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Número Máximo de Requisições Simultâneas	Número de requisições simultâneas que o cliente pode fazer ao servidor	1	1 a 8
Time-out de Comunicação	Time-out do nível da aplicação em ms	3000	10 a 65535
Modo	Define quando a conexão com o servidor é finalizada pelo cliente	Conexão é fechada após um tempo de inatividade de (s): 10 a 3600.	- Conexão é fechada depois de um time-out. - Conexão é fechada ao final de cada comunicação - Conexão é fechada após um tempo de inatividade de (s): 10 a 3600.
Tempo de Inatividade	Tempo de inatividade	10	3600

Tabela 4-58. Configurações Avançadas MODBUS Cliente

Notas:

Número Máximo de Requisições Simultâneas: É utilizado em servidores com um alto ciclo de varredura. Esse parâmetro é fixado em 1 (não editável), quando o protocolo configurado é MODBUS RTU via TCP.

Time-out de Comunicação: O time-out da comunicação é o tempo que o cliente aguardará por uma resposta do servidor à requisição. Para um dispositivo MODBUS Cliente, duas variáveis do sistema devem ser consideradas: o tempo que o servidor leva para processar a requisição e o atraso de envio da resposta caso seja configurado no servidor. É recomendado que o time-out seja igual ou maior que duas vezes a soma destes parâmetros. Para mais informações, ver capítulo Configuração das Interfaces Ethernet.

Modo: Define quando a conexão com o servidor é finalizada pelo cliente.

Tempo de Inatividade: Tempo de inatividade da conexão

ATENÇÂO: Caso o usuário utilize uma variável que abranja o endereço 5 a 10 de uma função declarada na aba de Mapeamentos, será válida a leitura/escrita de Requisições que atinjam as extremidades ou parte do Mapeamento. Exemplo, a leitura/escrita dessa função do endereço 1 ao 5, onde somente o endereço 5 está declarado, este endereço receberá valores válidos e o restante será desconsiderado, sem valores válidos.

Configuração dos Mapeamentos

A configuração das relações MODBUS Cliente, visualizadas na Figura 4-20, segue os parâmetros descritos na Tabela 4-59:

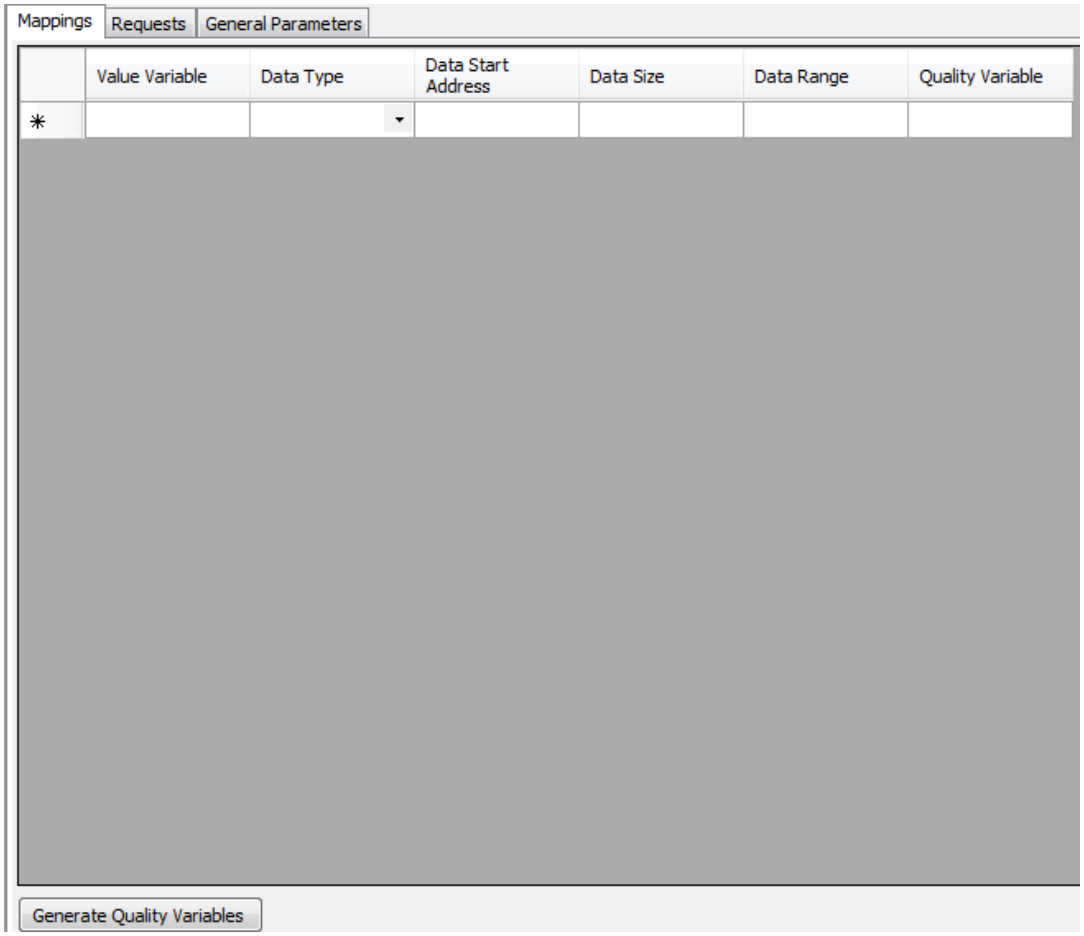


Figura 4-20 Tela de Mapeamentos de Dados MODBUS Cliente

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Value Variable	Nome da variável simbólica	-	Nome de uma variável declarada em um programa ou GVL
Data Type	Tipo de dado MODBUS	-	Coil Escrita(1 bit) Coil Leitura (1 bit) Holding Register Escrita (16 bits) Holding Register Leitura (16 bits) Holding Register – Mascara AND (16 bits) Holding Register – Mascara OR (16 bits) Input Register (16 bits) Input Status (1 bit)
Data Start Address	Endereço inicial dos dados MODBUS	-	1 a 65536
Data Size	Tamanho do dado MODBUS	-	1 a 65536
Data Range	Faixa de endereços do dado configurado	-	-
Quality Variable	Nome da variável que vai conter os dados de qualidade do mapeamento MODBUS	-	Nome de uma variável declarada em um programa ou GVL. A variável deve ser do tipo DWORD
Generate Quality Variables	Gera automaticamente na GVL Qualities variáveis do tipo QUALITY para cada Variável de Valor declarada. Para mais informações acesse o capítulo Conversões de Qualidade.	-	-

Tabela 4-59. Configuração dos Mapeamentos MODBUS Cliente

Notas:

Value Variable: Esse campo é utilizado para especificar uma variável simbólica na relação MODBUS.

Data Type: Esse campo é utilizado para especificar o tipo de dado utilizado na relação MODBUS.

Tipo de Dado	Tamanho [bits]	Descrição
Coil Escrita	1	Saída digital de escrita.
Coil Leitura	1	Saída digital de leitura.
Holding Register de Escrita	16	Saída analógica de escrita.
Holding Register de Leitura	16	Saída analógica de leitura.
Holding Register com mascara AND	16	Saída analógica que pode ser lida ou escrita com máscara AND.
Holding Register com mascara OR	16	Saída analógica que pode ser lida ou escrita com máscara OR.
Input Register	16	Entrada analógica que pode ser apenas lida.
Input Status	1	Entrada digital que pode ser apenas lida.

Tabela 4-60. Tipos de Dados Suportados no MODBUS Cliente

Data Start Address: Endereço inicial do dado de um mapeamento MODBUS.

Data Size: O valor de Tamanho especifica a quantidade máxima de dados que uma relação MODBUS poderá acessar, a partir do endereço inicial. Sendo assim, para ler uma faixa de endereços contínua, é necessário que todos os endereços estejam declarados em uma única relação. Este campo varia de acordo com o tipo de dado MODBUS configurado.

Data Range: Este campo mostra ao usuário a faixa de endereços de memória utilizada pela relação MODBUS.

Quality Variable: Este campo é utilizado para especificar uma variável que vai conter os dados da qualidade da relação MODBUS.

Configuração das Requisições

A configuração das requisições MODBUS, visualizadas na Figura 4-21, segue os parâmetros descritos na Tabela 4-61:

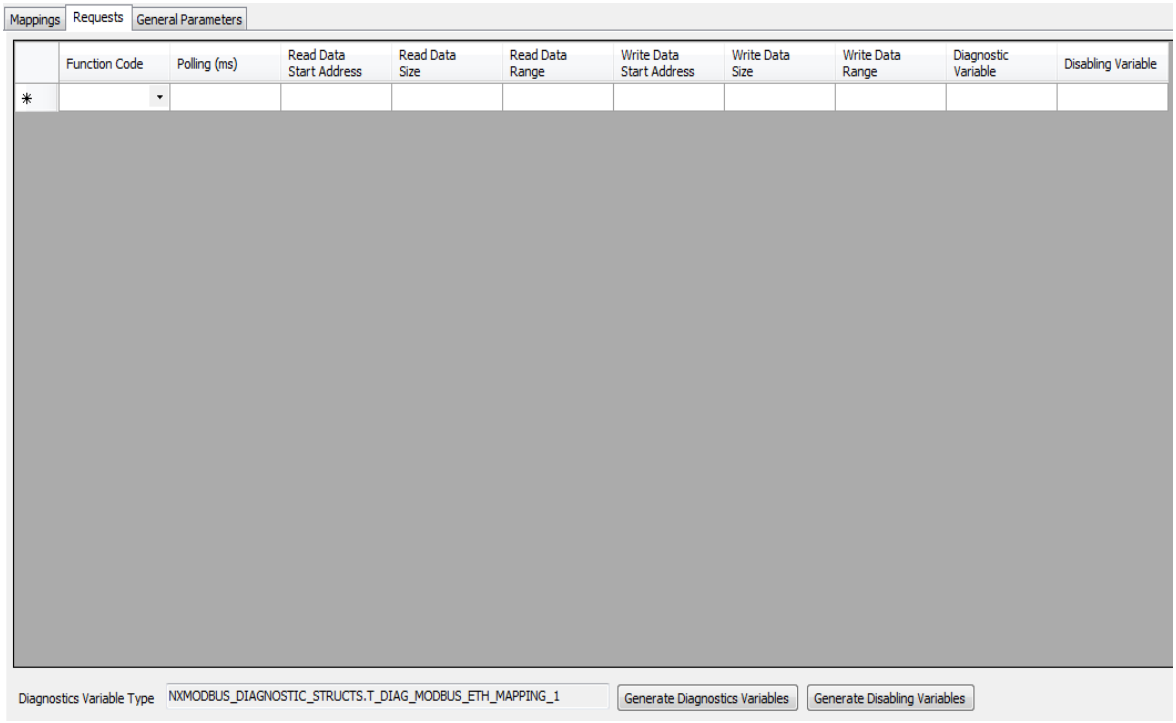


Figura 4-21. Tela de Requisições de dados MODBUS

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Function Code	Tipo de função MODBUS	-	FC01 – Leitura de Coils FC02 – Leitura de Input Status FC03 – Leitura de Holding Registers FC04 – Leitura de Input Registers FC05 – Escrita de um Coil FC06 – Escrita de um Holding Register FC15 – Escrita de múltiplos Coils FC16 – Escrita de Holding Registers FC22 – Máscara de Escrita do Register FC23 – Leitura/Escrita de Holding Register
Polling (ms)	Período de comunicação (ms)	-	0 a 3600000
Read Data Start Address	Endereço inicial dos dados de leitura MODBUS	-	1 a 65536
Read Data Size	Tamanho dos dados de leitura MODBUS	-	Depende da função utilizada
Read Data Range	Faixa de endereço dos dados de leitura MODBUS	-	0 a 2147483646
Write Data Start Address	Endereço inicial dos dados de escrita MODBUS	1	1 a 65536
Write Data Size	Tamanho dos dados de escrita MODBUS	-	Depende da função utilizada
Write Data Range	Faixa de endereço dos dados de escrita MODBUS	-	0 a 2147483647
Diagnostic Variable	Nome da variável de diagnóstico	-	Nome de uma variável declarada em um programa ou GVL
Disabling Variable	Variável utilizada para desabilitar a relação MODBUS	-	Campo destinado a variável simbólica utilizada para desabilitar, individualmente, as requisições MODBUS configuradas. Esta variável deve ser do tipo BOOL. A variável pode ser simples ou elemento de

			array e pode estar em estruturas.

Tabela 4-61. Configuração das Relações MODBUS Cliente

Notas:

Generate Diagnostic Variables: As variáveis simbólicas de diagnósticos podem ser geradas automaticamente através do botão "Gerar Variáveis de Diagnósticos". Ao clicar nesse botão, será criada uma variável do tipo" T_DIAG_MODBUS_ETH_MAPPING_1", na GVL "ReqDiagnostics", para cada uma das requisições.

Generate Disabling Variables: As variáveis simbólicas de desabilitação podem ser geradas automaticamente através do botão "Gerar Variáveis de Desabilitação". Ao clicar nesse botão, será criada uma variável do tipo BOOL, na GVL "Disables", para cada uma das requisições.

Configuração das relações MODBUS: O número de configurações, padrão de fábrica e os valores da coluna possibilidades, podem variar de acordo com o tipo de dado e função MODBUS (FC).

Function Code: As funções MODBUS (FC) disponíveis são os seguintes:

Tipo de Função	Código		Descrição
	DEC	HEX	
Acesso a Variáveis	1	0x01	Leitura de coils (FC 01)
	2	0x02	Leitura de input status (FC 02)
	3	0x03	Leitura de holding registers (FC 03)
	4	0x04	Leitura de input registers (FC 04)
	5	0x05	Escrita de um coil (FC 05)
	6	0x06	Escrita de um holding register (FC 06)
	15	0x0F	Escrita de múltiplos coils (FC 15)
	16	0x10	Escrita de holding registers (FC 16)
	22	0x16	Máscara de escrita do register (FC 22)
	23	0x17	Leitura/escrita holding registers (FC 23)

Tabela 4-62. Funções MODBUS Cliente

Polling: Este parâmetro indica com que frequência à comunicação definida por esta requisição deve ser executada. Ao ser finalizada uma comunicação será aguardado um tempo igual ao configurado no campo varredura e, após, será executada uma nova comunicação.

Read Data Start Address: Campo destinado ao endereço inicial dos dados de leitura MODBUS.

Read Data Size: O valor mínimo para o tamanho dos dados de leitura é 1 e o valor máximo depende da função MODBUS (FC) utilizada, conforme abaixo:

1. Leitura de Coils (FC 1): 2000
2. Leitura de Input Status (FC 2): 2000
3. Leitura de Holding Registers (FC 3): 125
4. Leitura de Input Registers (FC 4): 125
5. Leitura/Escrita de Holding Registers (FC 23): 121

Read Data Range: Este campo mostra a faixa de dados de leitura MODBUS configurada para cada requisição. O endereço inicial de leitura, somado ao tamanho do dado de leitura resultará na faixa de dados de leitura de cada uma das requisições.

Write Data Start Address: Campo destinado ao endereço inicial dos dados de escrita MODBUS.

Write Data Size: O valor mínimo para o tamanho dos dados de escrita é 1 e o valor máximo depende da função MODBUS (FC) utilizada, conforme abaixo:

1. Escrita de Um Coil (FC 5): 1
2. Escrita de Um Holding Registers (FC 6): 1
3. Escrita de Múltiplos Coils (FC 15): 1968
4. Escrita de Holding Registers (FC 16): 123
5. Máscara de Escrita do Register (FC 22): 1
6. Leitura/Escrita de Holding Registers (FC 23): 121

Write Data Range: Este campo mostra a faixa de dados de escrita MODBUS configurada para cada requisição. O endereço inicial de escrita, somado ao tamanho do dado de escrita resultará na faixa de dados de escrita de cada uma das requisições.

Diagnósticos MODBUS Cliente

Os diagnósticos da requisição MODBUS configurada, são armazenados em variáveis do tipo T_DIAG_MODBUS_ETH_CLIENT_1 a qual está descrita na Tabela 4-63 que podemos visualizar na Tabela 4-63.

Variável de diagnóstico do tipo T_DIAG_MODBUS_ETH_MAPPING_1.*		Tamanho	Descrição
Bits de estado da comunicação:			
byStatus.*	bCommIdle	BOOL	Comunicação inativa (aguardando ser constada).
	bCommExecuting	BOOL	Comunicação ativa.
	bCommPostponed	BOOL	Comunicação adiada, pois o número máximo de requisições simultâneas foi atingido. As comunicações adiadas serão executadas na mesma sequência em que elas foram requisitadas, evitando a indeterminação. O tempo gasto nesse estado não é contabilizado para efeitos de time-out. Os bits bCommIdle e bCommExecuting são falsos quando o bit bCommPostponed é verdadeiro.
	bCommDisabled	BOOL	Comunicação desabilitada. O bit bCommIdle é reiniciado nessa condição.
	bCommOk	BOOL	Comunicação finalizada anteriormente foi realizada com sucesso.
	bCommError	BOOL	Comunicação finalizada anteriormente teve um erro. Verificar código de erro.
	bDiag_6_reserved	BOOL	Reservado
	bDiag_7_reserved	BOOL	Reservado
Último código de erro (habilitado quando o bCommError = verdadeiro):			
eLastErrorCode		MASTER_ERROR_CODE (BYTE)	ERR_EXCEPTION (1): resposta reportada em um código de exceção (ver eLastExceptionCode = Código de Exceção). ERR_CRC (2): resposta com CRC inválido. ERR_ADDRESS (3): endereço MODBUS não encontrado. O endereço que respondeu à requisição foi diferente do esperado. ERR_FUNCTION (4): código inválido da função. A função recebida na resposta foi diferente da esperada pela requisição. ERR_FRAME_DATA_COUNT (5): a quantidade de dados da resposta foi diferente do esperado. ERR_INVALID_PROTOCOL_ID (6): protocolo não identificado. O protocolo da resposta é diferente do esperado. ERR_NOT_ECHO (7): resposta não é eco da pergunta (FC 5 e 6). ERR_REFERENCE_NUMBER (8): número de referência inválido (FC 15 e 16). ERR_INVALID_FRAME_SIZE (9): resposta menor do que a esperada. ERR_CONNECTION (20): erro durante a fase de conexão. ERR_SEND (21): erro durante a fase de

		transmissão. ERR_RECEIVE (22): erro durante a fase de recepção. ERR_CONNECTION_TIMEOUT (40): time-out no nível de aplicação durante a fase de conexão. ERR_SEND_TIMEOUT (41): time-out no nível de aplicação durante a transmissão. ERR_RECEIVE_TIMEOUT (42): time-out no nível de aplicação enquanto aguarda resposta. NO_ERROR (128): sem erro desde a inicialização.
Último código de exceção recebido pelo mestre:		
eLastExceptionCode	MODBUS_EXCEPTION (BYTE)	NO_EXCEPTION (0) FUNCTION_NOT_SUPPORTED (1) MAPPING_NOT_FOUND (2) ILLEGAL_VALUE (3) ACCESS_DENIED (128)* MAPPING_DISABLED (129)* IGNORE_FRAME (255)*
Estatísticas de comunicação:		
byDiag_3_reserved	BYTE	Reservado.
wCommCounter	WORD	Contador de comunicações finalizadas, com ou sem erros. O usuário pode testar quando a comunicação foi finalizada testando a variação desse contador. Quando o valor 65535 é atingido, o contador retorna a zero.
wCommErrorCounter	WORD	Contador de comunicações finalizadas com erros. Quando o valor 65535 é atingido, o contador retorna a zero.

Tabela 4-63. Diagnósticos das Relações MODBUS Cliente

Códigos de exceção: As definições dos códigos de exceção 128, 129 e 255 são válidas apenas na utilização de escravos Altus. Para escravos de outros fabricantes esses códigos de exceção podem ter significados diferentes.

Disabling Variable: Campo destinado a variável utilizada para desabilitar, individualmente, as requisições MODBUS configuradas no espaço Requisições. A requisição é desabilitada quando a variável, correspondente a requisição, for igual a 1, caso contrário, a requisição está habilitada.

ATENÇÃO:
Diferentemente de outras tarefas de uma aplicação, quando for atingida uma marca de depuração na MainTask, a tarefa de uma instância MODBUS Ethernet Cliente, e qualquer outra tarefa MODBUS, irá parar de ser executada no momento em que tentar efetuar uma escrita em uma área de memória. Isto ocorre para manter a consistência dos dados das áreas de memória enquanto a MainTask não estiver em execução.

Disparo de Relações MODBUS Cliente de Forma Acíclica

Para disparar relações MODBUS Cliente de forma acíclica, sugere-se o seguinte método, que pode ser implementado de maneira simples no programa aplicativo do usuário:

- Definir tempo máximo de polling para as relações
- Manter a relação normalmente desabilitada
- Habilitar a relação no momento em que se deseja executá-la
- Esperar pela confirmação de término da execução da relação, e neste momento desabilitá-la novamente

MODBUS Ethernet Servidor

Este protocolo está disponível para a UCP da Série Hadron Xtorm nos seus canais Ethernet. Ao selecionar esta opção no MasterTool Xtorm, a UCP passa a ser servidor da comunicação MODBUS, permitindo a conexão com dispositivos cliente MODBUS. Este protocolo somente está disponível quando a UCP estiver em modo de execução (Modo Run).

Para configurar este protocolo, é necessário executar os seguintes passos:

- Adicionar a instância do protocolo MODBUS Servidor a um dos canais Ethernet disponíveis (NET 1 ao NET 6)
- Para realizar esse procedimento, consultar o capítulo Inserindo uma Instância de Protocolo
- Configurar a interface Ethernet. Para realizar esse procedimento, consultar o capítulo Configuração das Interfaces Ethernet
- Configurar os parâmetros gerais do protocolo MODBUS servidor, como: porta TCP, seleção de protocolo, filtros de IP para Escrita e para Leitura (disponível no botão de configuração de filtros) e tempos de comunicação (disponível no botão de configurações avançadas do Servidor)
- Adicionar e configurar os mapeamentos MODBUS, especificando o nome da variável, tipo de dados, endereço inicial do dado e tamanho do dado

As descrições de cada configuração estão relacionadas a seguir, neste capítulo.

Parâmetros Gerais do Protocolo MODBUS Servidor

Os parâmetros gerais, encontrados na tela inicial de configuração do protocolo MODBUS (Figura 4-22), são definidos como:

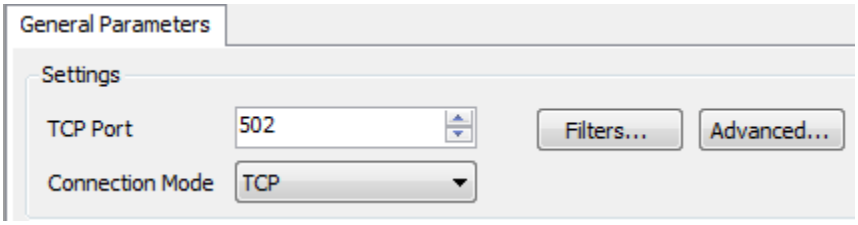


Figura 4-22. Tela de Configuração Parâmetros Gerais MODBUS Servidor

Configurações	Descrição	Padrão de Fábrica	Possibilidades
TCP Port	Porta TCP	502	2 a 65534
Connection Mode	Seleção do protocolo	TCP	RTU via TCP TCP

Tabela 4-64. Configurações Gerais MODBUS Servidor

Notas:

TCP Port: Caso sejam adicionadas várias instâncias do protocolo em uma única interface Ethernet, diferentes portas TCP devem ser selecionadas para cada instância. Algumas portas TCP, são reservadas, veja capítulo Portas TCP Reservadas.

- Filters

As configurações presentes no botão “Filters...”, descritas na Tabela 4-65, são relativas aos filtros de comunicação TCP:

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Write Filter IP Address	Especifica um intervalo de IPs com acesso de escrita nas variáveis declaradas na relação	0.0.0.0	0.0.0.0 a 255.255.255.255

	MODBUS		
Write Filter Mask	Especifica a máscara de subrede em conjunto com o parâmetro Filtro de IP para Escrita	0.0.0.0	0.0.0.0 a 255.255.255.255
Read Filter IP Address	Especifica um intervalo de IPs com acesso de leitura nas variáveis declaradas na relação MODBUS	0.0.0.0	0.0.0.0 a 255.255.255.255
Read Filter Mask	Especifica a máscara de subrede em conjunto com o parâmetro Filtro de IP para Leitura	0.0.0.0	0.0.0.0 a 255.255.255.255

Tabela 4-65. Filtros de IP

Nota:

Filtros: Os filtros são utilizados para estabelecer um intervalo de endereços IP que têm acesso de escrita ou leitura nas relações MODBUS, sendo individualmente configurados. O critério de permissão é realizado através de uma operação lógica AND entre o Filtro de Máscara para Escrita e o endereço IP do cliente. Caso o resultado seja igual ao Filtro de IP para Escrita, o cliente tem direito de escrita. Por exemplo, se o Filtro de IP para Escrita = 192.168.15.0 e o Filtro de Máscara para Escrita = 255.255.255.0, então somente clientes com endereço IP = 192.168.15.x terão direito de escrita. O mesmo procedimento é aplicado nos parâmetros de Filtro de Leitura para definir os direitos de leitura.

- Tempos de comunicação:

Os tempos de comunicação do protocolo MODBUS Servidor, encontrados no botão "Avançado..." da tela de configuração, estão divididos em: Ciclo da Tarefa e Time-out da Inatividade da Conexão.

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Ciclo da Tarefa (ms)	Tempo para execução da instância dentro do ciclo, sem considerar o tempo de execução da mesma	50	5 a 100
Time-out da Inatividade da Conexão (s)	Tempo máximo de inatividade entre cliente e servidor antes da conexão ser fechada pelo servidor	10	10 a 3600

Tabela 4-66. Configurações Avançadas

Notas:

Ciclo da Tarefa: O usuário deverá ter cuidado a alterar esse parâmetro, pois o mesmo interfere diretamente no tempo de resposta, volume de dados por varredura e, principalmente, no balanceamento dos recursos da UCP entre comunicações e outras tarefas.

Time-out da Inatividade da Conexão: Esse parâmetro foi criado para evitar que a quantidade máxima de conexões TCP seja atingida, imaginando que conexões inativas permanecessem abertas pelos mais diversos problemas. Enfim, indica por quanto tempo uma conexão (cliente ou servidora) pode permanecer aberta sem ser utilizada, ou seja, sem trocar mensagens de comunicação. Se o tempo especificado for atingido, a conexão simplesmente é fechada, liberando uma entrada na tabela de conexões.

Diagnósticos MODBUS Servidor

Os diagnósticos e comandos do protocolo MODBUS TCP Servidor configurado, são armazenados em variáveis do tipo T_DIAG_MODBUS_ETH_SERVER_1 as quais estão descritas na Tabela 4-67:

Variável de diagnósticos do tipo T_DIAG_MODBUS_ETH_SERVER_1 *.	Tamanho	Descrição
Bits de diagnóstico:		

tDiag.*	bRunning	BOOL	O servidor está em execução.
	bNotRunning	BOOL	O servidor não está em execução (ver bit bInterruptedByCommand).
	bInterruptedByCommand	BOOL	O bit bNotRunning foi habilitado, pois o servidor foi interrompido pelo usuário através de bits de comando.
	bConfigFailure	BOOL	Diagnóstico descontinuado.
	bRXFailure	BOOL	Diagnóstico descontinuado.
	bTXFailure	BOOL	Diagnóstico descontinuado.
	bModuleFailure	BOOL	Diagnóstico descontinuado.
	bDiag_7_reserved	BOOL	Reservado
byDiag_1_reserved		BYTE	Reservado
Bits de comando, reiniciados automaticamente:			
tCommand.*	bStop	BOOL	Parar o servidor.
	bRestart	BOOL	Reiniciar o servidor.
	bResetCounter	BOOL	Reiniciar a estatística dos diagnósticos (contadores).
	bDiag_19_reserved	BOOL	Reservado
	bDiag_20_reserved	BOOL	Reservado
	bDiag_21_reserved	BOOL	Reservado
	bDiag_22_reserved	BOOL	Reservado
	bDiag_23_reserved	BOOL	Reservado
byDiag_03_reserved		BYTE	Reservado
Estatísticas de comunicação:			
tStat.*	wActiveConnections	WORD	Número de conexões estabelecidas entre cliente e servidor (0 a 64).
	wTimeoutClosedConnections	WORD	Contador de conexões, entre cliente e servidor, interrompidas depois de um período de inatividade – time-out (0 a 65535).
	wClientClosedConnections	WORD	Contador de conexões interrompidas devido à solicitação do cliente (0 a 65535).
	wRXFrames	WORD	Contador de frames Ethernet recebidos pelo servidor, sendo que um frame Ethernet pode conter mais de uma requisição (0 a 65535).
	wRXRequests	WORD	Contador de requisições recebidas pelo servidor e respondidas normalmente (0 a 65535).
	wTXExceptionResponses	WORD	Contador de requisições recebidas pelo servidor e respondidas com códigos de exceção (0 a 65535). Os códigos de exceção estão relacionados abaixo: 1: o código da função (FC) é legal, porém não suportado. 2: relação não encontrada nestes dados MODBUS. 3: valor ilegal para o endereço. 128: o mestre/cliente não tem direito de escrita ou leitura. 129: a relação MODBUS está desabilitada.
	wRXIllegalRequests	WORD	Contador de requisições ilegais. (0 a 65535).
	wDiag_18_reserved	WORD	Reservado.

Tabela 4-67. Diagnósticos MODBUS Servidor

Nota:

Contadores: Todos os contadores dos diagnósticos do MODBUS Ethernet Servidor retornam a zero quando o valor limite 65535 é ultrapassado.

Configuração dos Mapeamentos

A configuração das relações MODBUS Servidor, visualizadas na Figura 4-23, segue os parâmetros descritos na Tabela 4-68:

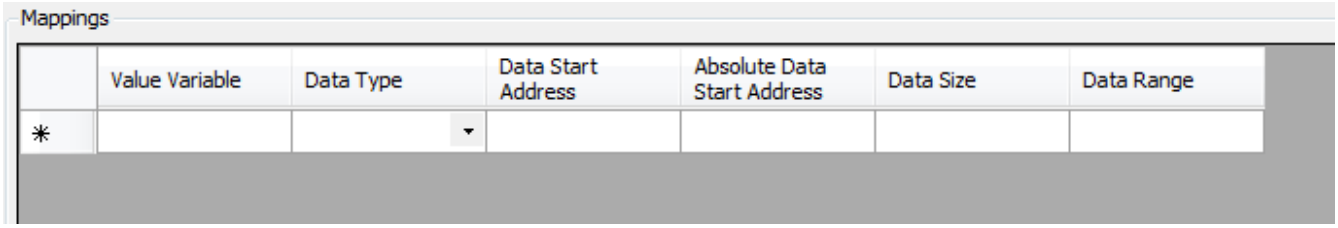


Figura 4-23. Tela de Mapeamentos de Dados MODBUS Servidor

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Value Variable	Nome da variável simbólica	-	Nome de uma variável declarada em um programa ou GVL
DataType	Tipo de dado MODBUS	-	Coil Input Status Holding Register Input Register
Data Start Address	Endereço inicial dos dados MODBUS	-	1 a 65536
Absolute Data Start Address	Endereço inicial absoluto dos dados MODBUS conforme o seu tipo.	-	-
Data Size	Tamanho do dado MODBUS	-	1 a 65536
Data Range	Faixa de endereços do dado configurado	-	-

Tabela 4-68. Configuração dos Mapeamentos MODBUS Ethernet

Notas:

Value Variable: Esse campo é utilizado para especificar uma variável simbólica na relação MODBUS.

Data Type: Esse campo é utilizado para especificar o tipo de dado utilizado na relação MODBUS.

Data Start Address: Endereço inicial do dado de um mapeamento MODBUS.

Absolute Data Start Address: Endereço inicial absoluto dos dados MODBUS conforme o seu tipo. Por exemplo, o Holding Register com endereço 5 possui endereço absoluto 400005. Este é campo é apenas leitura e está disponível para auxiliar na configuração do Client/Master MODBUS que irá comunicar-se com este dispositivo. Os valores dependem do endereço base (offset) de cada tipo de dado MODBUS e do endereço permitido para cada tipo de dado.

Data Size: O valor de Tamanho especifica a quantidade máxima de dados que uma relação MODBUS poderá acessar, a partir do endereço inicial. Sendo assim, para ler uma faixa de endereços contínua, é necessário que todos os endereços estejam declarados em uma única relação. Este campo varia de acordo com o tipo de dado MODBUS configurado.

Data Range: É um campo somente de leitura e informa a faixa de endereços que está sendo usada por esse mapeamento. Ele é formado pela soma dos campos "Endereço Inicial" e "Tamanho". Não podem haver sobreposições de faixa com outros mapeamentos do mesmo "Tipo de Dado".

Máximo Tamanho de Relação: Cada NET tem limite de 20000 pontos de comunicação, ou seja, caso o usuário utilize os 20000 pontos em uma única relação na NET1, não será possível utilizar mais pontos de comunicação nesta NET. Entretanto uma boa prática é utilizar quantidades e tipos de relações apropriadas às necessidades da aplicação. Para mais informações ver capítulo Configuração das Interfaces Ethernet.

ATENÇÃO: Diferentemente de outras tarefas de uma aplicação, quando for atingida uma marca de depuração na MainTask, a tarefa de uma instância MODBUS Ethernet Servidor, e qualquer outra tarefa MODBUS, irá parar de ser executada no momento em que tentar efetuar uma escrita em uma área de memória. Isto ocorre para manter a consistência dos dados das áreas de memória enquanto a MainTask não estiver em execução.

Desempenho da Comunicação MODBUS

Os dispositivos MODBUS configuráveis na UCP Hadron Xtorm executam em segundo plano, com uma prioridade abaixo da aplicação de usuário e de forma cíclica. Sendo assim, seu desempenho irá variar de acordo com o tempo restante, levando em consideração a diferença entre o intervalo e tempo que a aplicação leva para ser executada. Por exemplo, um dispositivo MODBUS em uma aplicação que é executada a cada 100 ms, com um tempo de execução de 50 ms, irá ter um desempenho menor do que com uma aplicação de 50 ms executando a cada 200 ms de intervalo. Isto acontece porque, no segundo caso, a UCP terá um tempo maior entre cada ciclo da MainTask para executar as tarefas com prioridade mais baixa.

Também deve-se levar em conta o número de ciclos que o dispositivo, escravo ou servidor, leva para responder a uma requisição.

Para processar e transmitir uma resposta, um MODBUS RTU Escravo irá levar dois ciclos (tempo do ciclo da tarefa MODBUS), enquanto que um MODBUS Ethernet Servidor levará apenas um ciclo. Porém, esse é o tempo mínimo entre a recepção de uma requisição e o envio da resposta. Caso a requisição seja enviada logo após a execução de um ciclo da tarefa MODBUS, o tempo poderá ser equivalente a 2 ou 3 vezes o tempo de ciclo para o MODBUS Escravo, e de 1 a 2 vezes o tempo de ciclo para o MODBUS Servidor.

Neste caso: Tempo Máximo de Resposta = 3*(tempo de ciclo) + (tempo de execução das tarefas) + tempo interframe chars) + (send delay).

Por exemplo, para uma tarefa MODBUS Ethernet Servidor com um ciclo de 50 ms, em uma aplicação executada por 60 ms a cada 100 ms, o servidor conseguirá executar apenas um ciclo entre cada ciclo da aplicação. Por outro lado, com a mesma aplicação, sendo executada por 60 ms, mas com um intervalo de 500 ms, o MODBUS terá um desempenho melhor, pois enquanto a aplicação não estiver em execução, o mesmo estará sendo executado a cada 50 ms e apenas a cada ciclo da MainTask ele irá demorar mais para ser executado. Para estes casos, o pior desempenho será a soma do tempo de execução da aplicação do usuário com o tempo do ciclo da tarefa MODBUS.

Para os dispositivos MODBUS RTU Mestre e MODBUS Ethernet Cliente o princípio de funcionamento é exatamente igual, mas leva em consideração o tempo de varredura (ms) configurado na relação MODBUS e não o tempo do ciclo da tarefa MODBUS. Para esses casos, o tempo de ciclo da tarefa MODBUS é especificado da seguinte maneira:

MODBUS RTU Mestre

Tempo de ciclo da tarefa = (Menor Tempo de Varredura (ms) configurado nas relações / 5)

Caso o tempo de ciclo da tarefa, calculado de acordo com a expressão acima, fique maior que o valor configurado no campo “Time-out de Comunicação” disponível na aba de Configurações Avançadas do Dispositivo, o Tempo de ciclo da tarefa será igual:

Tempo de ciclo da tarefa = Time-out de Comunicação (ms) configurado no dispositivo

MODBUS Ethernet Cliente

Tempo de ciclo da tarefa = (Menor Tempo de Varredura (ms) configurado nas relações / 3)

Caso o tempo de ciclo da tarefa, calculado de acordo com a expressão acima, fique maior que o valor configurado no campo “Time-out de Comunicação” disponível na aba de Configurações Avançadas do Dispositivo, o Tempo de ciclo da tarefa será igual:

Tempo de ciclo da tarefa = Time-out de Comunicação (ms) configurado no dispositivo

Para estes casos, o pior desempenho de uma relação será ser executada após o seu tempo de varredura, somado ao tempo de execução da aplicação de usuário.

É importante ressaltar que o número de dispositivos MODBUS em execução também irá alterar o seu desempenho. Em uma aplicação de usuário com tempo de execução de 60 ms e intervalo de 100 ms, irá restar 40 ms para a UCP executar todas as tarefas de menor prioridade. Portanto, uma UCP com apenas um MODBUS Ethernet Servidor terá um desempenho maior do que uma UCP que utilize quatro destes dispositivos.

Taxa de Comunicação de um Dispositivo MODBUS Servidor

Para um dispositivo MODBUS Ethernet Servidor, podemos afirmar que ele é capaz de responder a um número x de requisições por segundo, ou seja, será capaz de transferir n bytes por segundo, dependendo do tamanho de cada requisição. Quanto menor for o ciclo da tarefa do Servidor MODBUS, maior será o impacto do número de conexões em sua taxa de resposta. Porém, para tempos de ciclo menores que 20 ms este impacto não é linear.

ATENÇÃO: Os desempenhos de comunicação mencionados neste capítulo são exemplos utilizando uma UCP apenas com um dispositivo MODBUS TCP Servidor, sem nenhuma lógica na aplicação que possa atrasar a comunicação. Portanto, estes desempenhos devem ser tomados como máximos.

Já para tempos de ciclo iguais ou maiores do que 20 ms, o crescimento da taxa de respostas é linear, podendo ser calculada através da fórmula:

N =- C x (Z – (Z / (T x 1000)))

Z = 1 / T

Onde “N” é o número médio de respostas por segundo, “C” é o número de conexões e T é igual ao tempo de ciclo da tarefa MODBUS (em segundos).

Tomando como exemplo um Servidor MODBUS com uma conexão e um tempo de ciclo de 50 ms temos:

C = 1; **T** = 0,05 s; **Z** = 1 / 0,05 = 20;

N = 1 x (20 – (20 / (0,05 x 1000))) = 1 x (20 – (20 / 50)) = 1 x (20 – 0,4) = 1 x 19,6

N = 19,6

Ou seja, nesta configuração o Servidor MODBUS será capaz de responder, em média, 19 requisições por segundo.

Caso este valor obtido seja multiplicado pelo número de bytes em cada requisição, será obtida uma taxa de transferência de n bytes por segundo.

DNP3 – Tipos de Dados

A Tabela 4-69 mostra o tipo de variável suportada pela HX3040 para cada um dos tipos de dados do protocolo DNP3.

Tipo de Mapeamento	Tipos de Variáveis IEC	Variação Válida de StaticGroup	Variação Válida de EventGroup	Event Class Faixa
BI: Binary Input	BOOL BIT	g01v01 - Entrada digital em formato compactado g01v02 - Entrada digital com qualidade	g02v01 - Evento de entrada digital sem tempo g02v02 - Evento de entrada digital com tempo absoluto g02v03 - Evento de entrada digital com tempo relativo	0..3
DBI: Double Binary Input	DBP	g03v01 - Entrada digital dupla em formato compactado g03v02 - Entrada digital dupla com qualidade	g04v01 - Evento de entrada digital dupla sem tempo g04v02 - Evento de entrada digital dupla com tempo absoluto g04v03 - Evento de entrada digital dupla com tempo relativo	0..3
BO: Binary Output	BOOL DBP BIT	g10v01 - Saída digital em formato compactado g10v02 - Saída digital com qualidade		0
CN: Counter	UINT	g20v02 - Contador de 16 bits com qualidade g20v06 - Contador de 16 bits sem qualidade	g22v02 - Evento do contador de 16 bits com qualidade g22v06 - Evento do contador de 16 bits com qualidade e tempo	0..3
	UDINT	g20v01 - Contador de 32 bits com qualidade g20v05 - Contador de 32 bits sem qualidade	g22v01 - Evento do contador de 32 bits com qualidade g22v05 - Evento do contador de 32 bits com qualidade e tempo	0..3
FCN: Frozen Counter	UINT	g21v02 - Contador congelado de 16 bits com qualidade g21v10 - Contador congelado de 16 bits sem qualidade	g23v02 - Evento do contador congelado de 16 bits com qualidade g23v06 - Evento do contador congelado de 16 bits com qualidade e tempo	0..3
	UDINT	g21v01 - Contador congelado de 32 bits com qualidade g21v09 - Contador congelado de 32 bits sem qualidade	g23v01 - Evento do contador congelado de 32 bits com qualidade g23v05 - Evento do contador congelado de 32 bits com qualidade e tempo	0..3
AI: Analog Input	INT	g30v02 - Entrada analógica de 16 bits com qualidade g30v04 - Entrada analógica de 16 bits sem qualidade	g32v02 - Evento de entrada analógica de 16 bits sem tempo g32v04 - Evento de entrada analógica de 16 bits com tempo	0..3
	DINT	g30v01 - Entrada analógica de 32 bits com qualidade g30v03 - Entrada analógica de 32 bits sem qualidade	g32v01 - Evento de entrada analógica de 32 bits sem tempo g32v03 - Evento de entrada analógica de 32 bits com tempo	0..3
	REAL	g30v05 - Entrada analógica de precisão simples com qualidade	g32v05 - Evento de entrada analógica de precisão simples sem tempo g32v07 - Evento de entrada analógica de precisão simples em ponto flutuante com tempo	0..3
AO: Analog Output	INT	g40v02 - Saída analógica de 16 bits com qualidade		0
	DINT	g40v01 - Saída analógica de 32 bits com qualidade		0
	REAL	g40v03 - Saída analógica de precisão simples com qualidade		0

Tabela 4-69. Declaração de Variáveis para DNP3

DNP3 Ethernet Cliente

Este protocolo está disponível para a UCP da Série Hadron Xtorm nos seus canais Ethernet. Ao selecionar esta opção no MasterTool Xtorm, a UCP passa a ser cliente da comunicação DNP3. O limite geral máximo de instanciamento de drivers de comunicação permitido para a UCP HX3040 é 12 drivers. Logo, o limite máximo de instanciamento de drivers Cliente DNP3, depende também de quantos drivers de outros protocolos estão instanciados. Este protocolo somente está disponível quando a UCP estiver em modo de execução (Modo Run).

Para configurar este protocolo, é necessário executar os seguintes passos:

- Adicionar a instância do protocolo Cliente DNP3 a um dos canais Ethernet disponíveis (NET 1 ao NET 6)
- Para realizar esse procedimento, consultar o capítulo Inserindo uma Instância de Protocolo
- Configurar a interface Ethernet. Para realizar esse procedimento, consultar o capítulo Configuração das Interfaces Ethernet
- Configurar os parâmetros gerais do protocolo Cliente DNP3, com o modo Endereço de Enlace
- Adicionar e configurar os dispositivos outstation, definindo os devidos parâmetros. Para cada driver instanciado, existe um limite máximo de 32 outstations
- Adicionar e configurar os mapeamentos DNP3, especificando o nome da variável, grupo estático, índices, tamanho, faixa, classe e qualidade
- Configurar os parâmetros da camada de enlace, especificando os endereçamentos e as mensagens de confirmação
- Configurar os parâmetros da camada de aplicação, especificando as configurações da camada de aplicação, varredura das classes e solicitações automáticas

As descrições de cada configuração estão relacionadas a seguir, neste capítulo.

Configuração dos Mapeamentos Cliente DNP3

As configurações dos Parâmetros Gerais de um Cliente DNP3 conforme a Figura 4-24 seguem os parâmetros da Tabela 4-70 apresentadas a seguir:

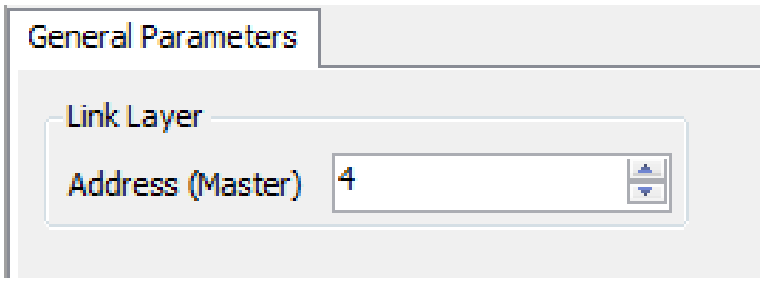


Figura 4-24. Tela de Parâmetros Gerais do Cliente DNP3

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Address (Master)	Define o endereço do mestre DNP3.	4	0 a 65519

Tabela 4-70. Configuração dos Parâmetros Gerais do Cliente DNP3

A configuração das relações Cliente DNP3, visualizadas na Figura 4-25, seguem os parâmetros descritos na Tabela 4-71:

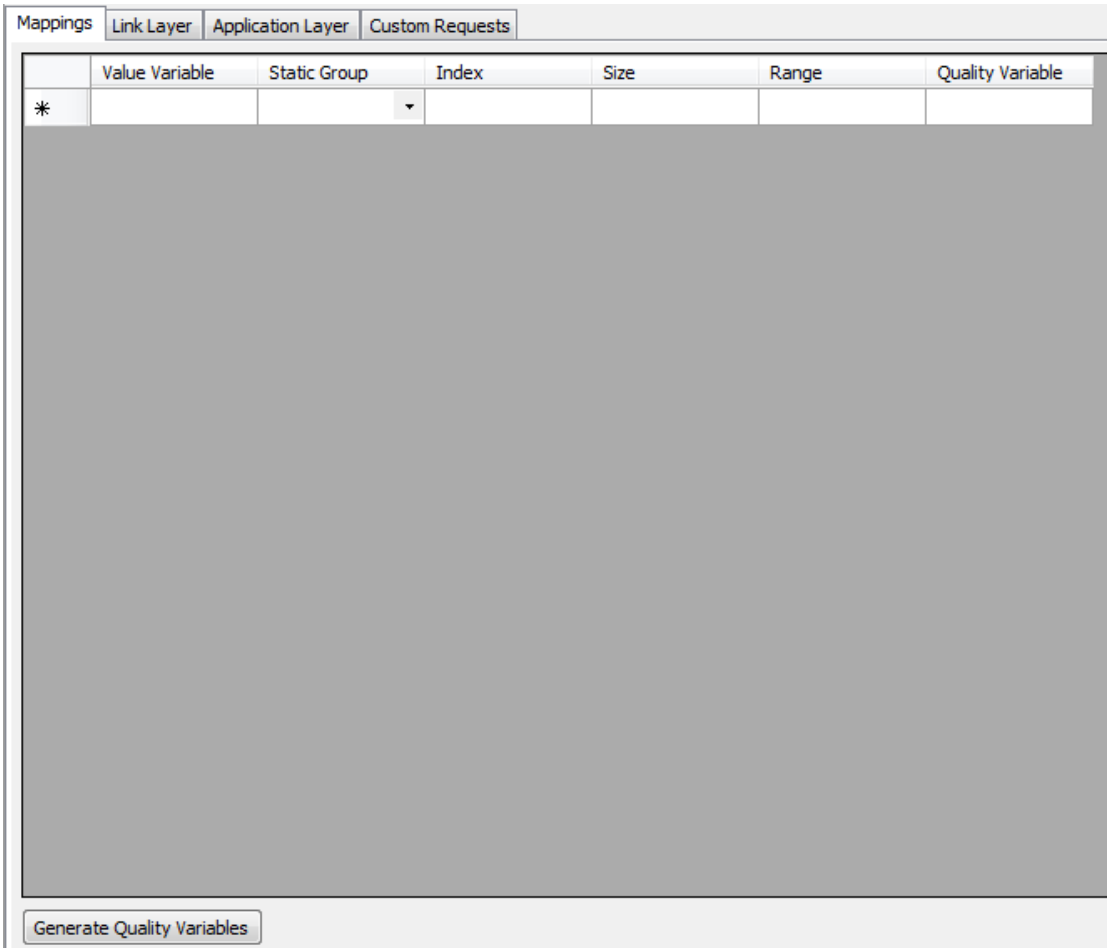


Figura 4-25. Tela de Mapeamentos de Dados Cliente DNP3

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Value Variable	Nome da variável simbólica	-	Nome de uma variável declarada em um programa ou GVL
Static Group	Configuração dos dados estáticos	-	g01 – Entrada Digital g02 – Entrada Digital Dupla g10 – Saída Digital g20 – Contador g21 – Contador Congelado g30 – Entrada Analógica* g40 – Saída analógica*
Index	Índice do primeiro ponto de mapeamento DNP3	-	0 a 65535
Size	Tamanho do dado DNP3	-	1 a 65536
Range	Faixa de endereços do dado configurado	-	-
Quality Variable	Nome da variável simbólica que armazenará a qualidade do mapeamento	-	Variável do tipo QUALITY
Generate Quality Variable	Gera automaticamente na GVL Qualities variáveis do tipo QUALITY para cada Variável de Valor declarada. Para mais informações acesse o capítulo Conversões de Qualidade.	-	-

Tabela 4-71. Configuração dos Mapeamentos Cliente DNP3

Notas:

Value Variable: Nome da variável simbólica a ser mapeada, Quando um comando de leitura é enviado, o retorno enviado na resposta será armazenado nesta variável. Quando for um comando de escrita, o valor escrito será armazenado nesta variável. A variável pode ser simples, array ou elemento de array e pode estar em estruturas.

Static Group: É utilizado para selecionar em qual grupo DNP3 que o mapeamento será criado.

Index: O campo é utilizado para configurar o índice do primeiro ponto do mapeamento no protocolo DNP3.

Size: É um campo somente leitura que informa quantos pontos estão sendo usados por esse mapeamento. O valor deste campo varia conforme o tipo da variável configurada no mapeamento.

Range: É um campo somente de leitura que informa a faixa de endereços que está sendo usada por esse mapeamento. Ele é formado pela soma dos campos "Índices" e " Tamanho". Não pode haver sobreposições de faixa com outros mapeamentos do mesmo tipo.

Quality Variable: Nome da variável simbólica que armazenará a qualidade do mapeamento, sendo esta, sempre do tipo QUALITY, disponível a partir da biblioteca LibDataTypes. A variável pode ser simples, array ou elemento de array e pode estar em estruturas.

ATENÇÃO:
Quando variáveis do grupo 30 (Analógicas de entrada) e grupo 40 (Analógicas de saída) recebem valores do Outstation que não podem ser armazenados devido à diferença entre as variações dos dados recebidos e da variável de armazenamento mapeada no Cliente DNP3, a qualidade daquele ponto será INVALID e o valor do ponto será o último valor válido lido pelo Cliente DNP3. Isto ocorrerá somente no caso do Servidor DNP3 responder em uma variação maior do que a mapeada no Cliente DNP3. No caso do Servidor DNP3 responder na mesma variação ou em uma variação inferior, o valor e a qualidade do ponto serão armazenados conforme a resposta do Servidor DNP3. Por exemplo: Variável do tipo INT (16-bit) mapeada no Cliente DNP3 e Servidor DNP3 responde na variação de 32-bit. Neste caso, o valor contido na resposta do Servidor DNP3 não será atualizado e a qualidade será INVALID.

ATENÇÃO:
O driver DNP3 Client não gera eventos baseados nos valores de seus mapeamentos. Os eventos devem ser gerados pelo respectivo IED. Sendo assim, caso o IED não tenha suporte a geração de eventos, o driver Cliente DNP3 não será capaz de gerá-los.

Configuração da Camada de Enlace Cliente DNP3

A configuração dos parâmetros da camada de enlace do Cliente DNP3, visualizadas na Figura 4-31, segue os parâmetros descritos na Tabela 4-72:

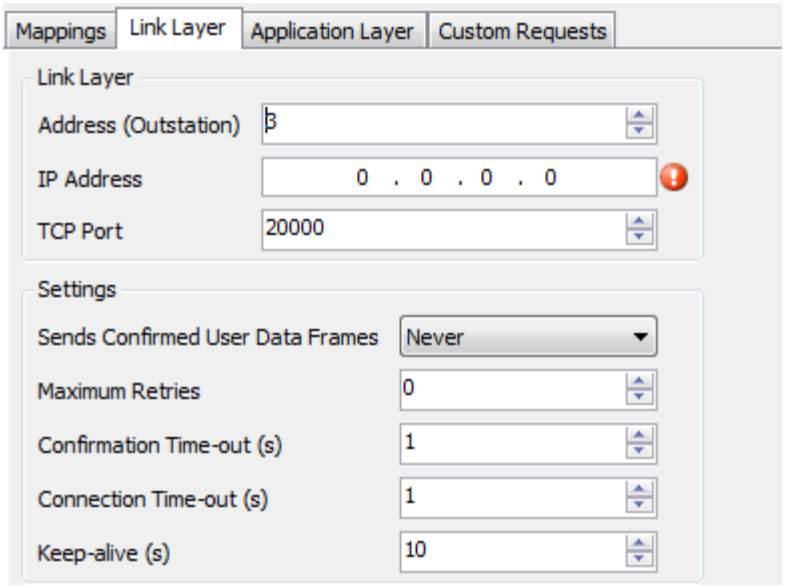


Figura 4-26. Tela de Configurações da Camada de Enlace Cliente DNP3

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Address (Outstation)	Endereço DNP3 de origem deste Outstation	3	0 a 65519
IP Address	Endereço IP deste Outstation	0.0.0.0	1.0.0.1 a 223.255.255.254
TCP Port	Endereço da porta listen para conexão com este Outstation	20000	0 a 65535
Sends Confirmed User Data Frames	Seleciona o modo de confirmação de mensagens	Never	Never Sometimes Always
Maximum Retries	Número de vezes que este Outstation irá retransmitir a mensagem caso ele não receba a confirmação do cliente.	0	0 a 10
Confirmation Time-Out(s)	Tempo para que este Outstation envie a mensagem de confirmação novamente	1	1 a 86400
Connection Time-Out(s)	Tempo máximo para que este Outstation se conecte com o DNP3 Cliente através do protocolo TCP.	1	1 a 10
Keep Alive (s)	Tempo do teste de link	10	1 a 60

Tabela 4-72. Configuração da Camada de Enlace Cliente DNP3

Notas:

Address (Outstation): O campo é utilizado para configurar o endereço DNP3 de origem deste outstation.

IP Address: O campo é utilizado para configurar o endereço IP do "Cliente DNP3".

Porta TCP: Endereço da porta listen para conexão com o "Cliente DNP3".

Sends Confirmed User Data Frames: Os modos de confirmação da camada de enlace disponíveis são os seguintes:

	Configurações	Descrição
Modo de Confirmação da Camada de Enlace	Nunca	Nunca solicitará confirmação das mensagens
	Sempre	Sempre solicitará confirmação das mensagens
	Ás vezes	Solicitará confirmação somente para mensagens multifragmentadas

Tabela 4-73. Modo de Confirmação da Camada de Enlace do Cliente DNP3

Maximum Retries: Número de vezes que o dispositivo “Cliente DNP3” irá retransmitir a mensagem caso não receba a confirmação do cliente. É habilitado quando, na opção “Modo de Confirmação da Camada de Enlace”, estiver selecionado “Sempre” ou “Às vezes”.

Confirmation Time-Out(s): Determina qual é o tempo (em segundos) para que o “Cliente DNP3” envie a mensagem de confirmação novamente. É habilitado quando, na opção “Modo de Confirmação da Camada de Enlace”, estiver selecionado “Sempre” ou “Às vezes”.

Keep Alive(s): Intervalo de tempo para envio de mensagens de teste de link ("TCP keep-alive" ou "serial link status") após última mensagem DNP recebida. Caso não seja recebida uma resposta para a mensagem de teste de link, a conexão atual será encerrada.

Configuração da Camada de Aplicação Cliente DNP3

A configuração das relações Cliente DNP3, visualizadas na Figura 4-27, segue os parâmetros descritos na Tabela 4-74:

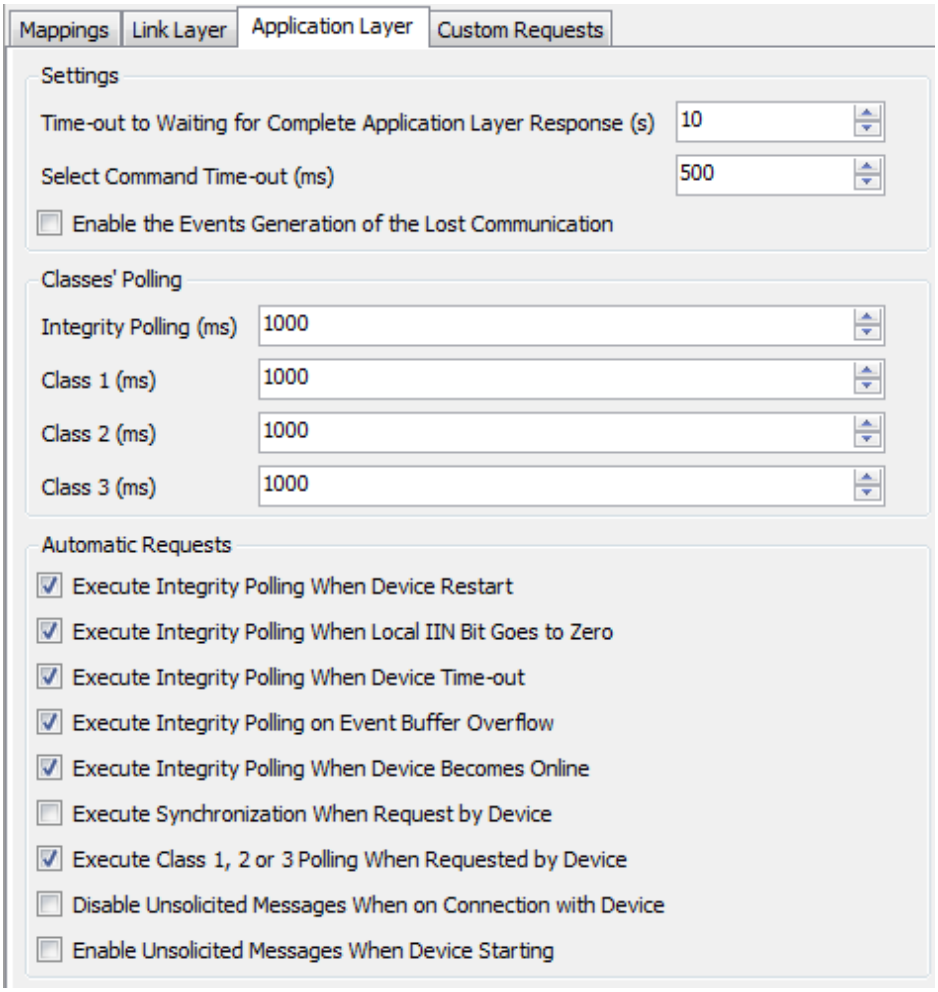


Figura 4-27. Tela de Configurações da Camada de Aplicação Cliente DNP3

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Time-out to Waiting for Complete Application Layer Response (s)	Tempo máximo que o outstation aguardará resposta completa da camada de aplicação	10	1 a 86400
Select Command Time-out (ms)	Tempo máximo para a execução do comando de	500	100 a 10000

	seleção		
Enable the Events Generation of the Lost Communication	Habilita a geração de eventos nas situações de perda de comunicação e restabelecimento de comunicação com o Outstation	Disabled	Enabled Disabled
Integrity Polling (ms)	Determina o tempo (em milissegundos) em que é executada a varredura de todos os pontos mapeados	1000	0 a 86400000
Class 1 (ms)	Tempo em que é executada a varredura dos pontos mapeados para a " Classe 1"	1000	0 a 86400000
Class 2 (ms)	Tempo em que é executada a varredura dos pontos mapeados para a " Classe 2"	1000	0 a 86400000
Class 3 (ms)	Tempo em que é executada a varredura dos pontos mapeados para a " Classe 3"	1000	0 a 86400000
Execute Integrity polling When Device Restart	Atualiza os valores de todos os pontos do dispositivo outstation e também lê todos os eventos quando o bit DEVICE_RESTART do IIN estiver em um.	Enabled	Enabled Disabled
Execute Integrity Polling When Local IIN Bit Goes to Zero	Executa a varredura de integridade quando o bit LOCAL do IIN passar de um para zero.	Enabled	Enabled Disabled
Execute Integrity Polling When Device Time-out	Executa a varredura de integridade quando o outstation estiver operando em modo local, não sendo acessível pelo mestre.	Enabled	Enabled Disabled
Execute Integrity Polling on Event Buffer Overflow	Executa a varredura de integridade após um estouro na fila de eventos do outstation, indicado pelo bit EVENT_BUFER_OVERFLOW do IIN.	Enabled	Enabled Disabled
Execute Integrity Polling When Device Becomes Online	Executa a varredura de integridade quando a comunicação com o mestre for restabelecida após um time-out.	Enabled	Enabled Disabled
Execute Synchronization When Request By Device	Permite que o mestre envie o comando de acerto de hora quando o outstation solicitar.	Disabled	Enabled Disabled
Execute Class 1,2 or 3 Polling When Requested by Device	Executa a leitura dos eventos das três classes quando um dos bits CLASS_1_EVENTS, CLASS_2_EVENTS e CLASS_3_EVENTS do IIN estiver em um.	Enabled	Enabled Disabled
Disable Unsolicited Messages When on Connection with Device	Envia o comando de desativação das mensagens não solicitadas quando o outstation conectar. Quando este campo é marcado, a caixa de seleção "Habilitar mensagens não solicitadas quando o dispositivo iniciar" é desmarcada e desabilitada.	Disabled	Enabled Disabled
Enable Unsolicited Messages When Device Starting	Envia o comando de ativação das mensagens não solicitadas quando o outstation conectar.	Disabled	Enabled Disabled

Tabela 4-74. Configuração da Camada de Aplicação Cliente DNP3

Requisições Customizadas Cliente DNP3

A configuração das relações Servidor DNP3, visualizadas na Figura 4-28, segue os parâmetros descritos na Tabela 4-75:

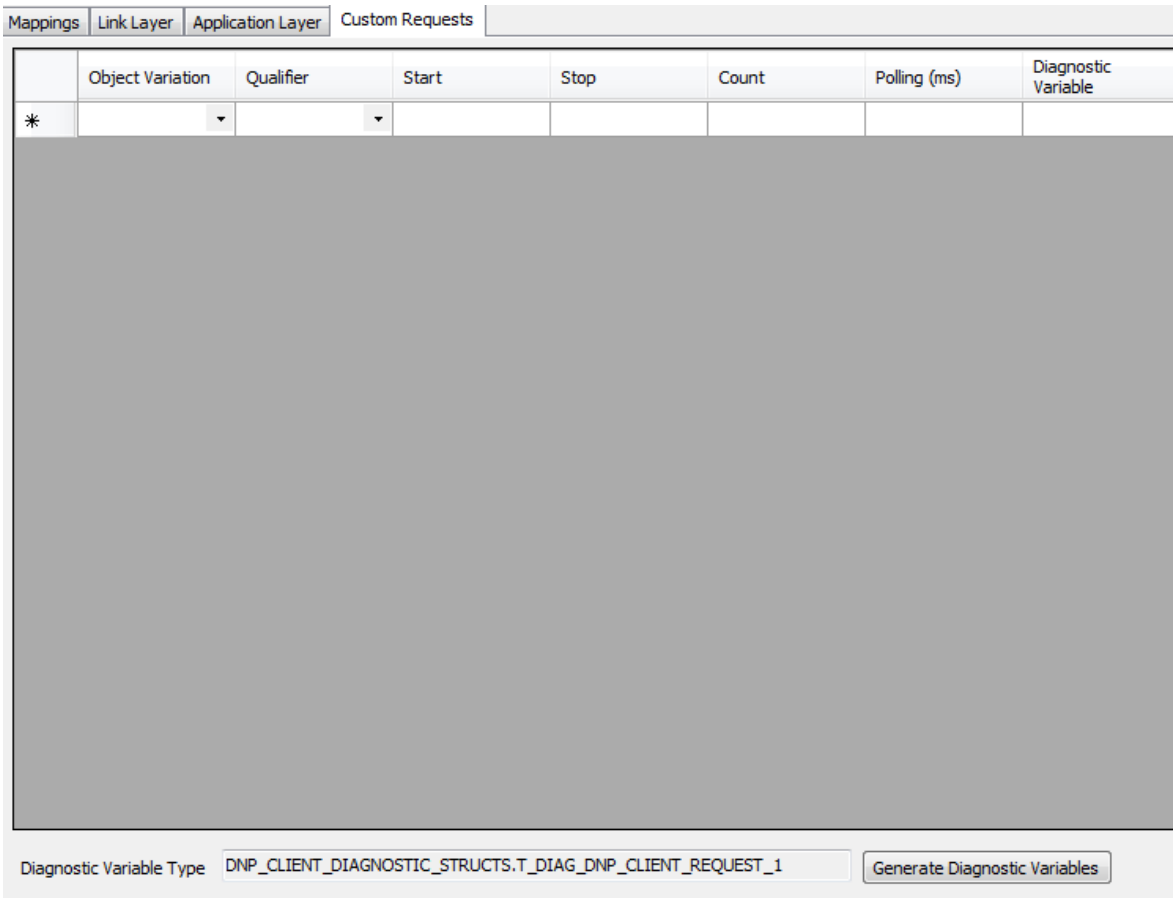


Figura 4-28. Tela de Mensagens Requisições Customizadas Cliente DNP3

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Object Variation	Define o grupo e a variação dos objetos que serão requisitados.	-	g01v00 - Entrada digital com variação. g01v01 - Entrada digital em formato compactado. g01v02 - Entrada digital com qualidade. g02v00 - Evento de entrada digital com variação. g02v01 - Evento de entrada digital sem tempo. g02v02 - Evento de entrada digital com tempo absoluto. g02v03 - Evento de entrada digital com tempo relativo. g03v00 - Entrada digital dupla com variação. g03v01 - Entrada digital dupla em formato compactado. g03v02 - Entrada digital dupla com qualidade. g04v00 - Evento de entrada digital dupla com variação. g04v01 - Evento de entrada digital dupla sem tempo.

			g04v02 - Evento de entrada digital dupla com tempo absoluto. g04v03 - Evento de entrada digital dupla com tempo relativo. g10v00 - Saída digital com variação. g10v01 - Saída digital em formato compactado. g10v02 - Saída digital com qualidade. g20v00 - Contador com variação. g20v01 - Contador de 32 bits com qualidade. g20v02 - Contador de 16 bits com qualidade. g20v05 - Contador de 32 bits sem qualidade. g20v06 - Contador de 16 bits sem qualidade. g21v00 - Contador congelado com variação. g21v01 - Contador congelado de 32 bits com qualidade. g21v02 - Contador congelado de 16 bits com qualidade. g21v09 - Contador congelado de 32 bits sem qualidade. g21v10 - Contador congelado de 16 bits sem qualidade. g22v00 - Evento do contador com variação. g22v01 - Evento do contador de 32 bits com qualidade. g22v02 - Evento do contador de 16 bits com qualidade. g22v05 - Evento do contador de 32 bits com qualidade e tempo. g22v06 - Evento do contador de 16 bits com qualidade e tempo. g23v00 - Evento do contador congelado com variação. g23v01 - Evento do contador congelado de 32 bits com qualidade. g23v02 - Evento do contador congelado de 16 bits com qualidade. g23v05 - Evento do contador congelado de 32 bits com qualidade e tempo. g23v06 - Evento do contador congelado de 16 bits com qualidade e tempo. g30v00 - Entrada analógica com variação. g30v01 - Entrada analógica de 32 bits com qualidade. g30v02 - Entrada analógica de 16 bits com qualidade. g30v03 - Entrada analógica de 32 bits sem qualidade. g30v04 - Entrada analógica de 16 bits sem qualidade. g30v05 - Entrada analógica de precisão simples com qualidade. g32v00 - Evento de entrada analógica com variação. g32v01 - Evento de entrada analógica de 32 bits sem tempo. g32v02 - Evento de entrada analógica de 16 bits sem tempo. g32v03 - Evento de entrada analógica de 32 bits com tempo. g32v04 - Evento de entrada analógica de 16 bits com tempo. g32v05 - Evento de entrada analógica de precisão simples sem tempo. g32v07 - Evento de entrada analógica de precisão simples em ponto flutuante com tempo. g40v00 - Saída analógica com variação. g40v01 - Saída analógica de 32 bits com qualidade. g40v02 - Saída analógica de 16 bits com qualidade.

			g40v03 - Saída analógica de precisão simples com qualidade. g60v01 - Classe de dados 0. g60v02 - Classe de dados 1. g60v03 - Classe de dados 2. g60v04 - Classe de dados 3.
Qualifier	Identifica qual o tipo de requisição que será realizada.	**-**	Faixa Todos Contador
Start	Define qual é o primeiro ponto de uma faixa que será requisitada. Só pode ser preenchido se a coluna "Qualificador" estiver com o valor "Faixa" selecionado.	0	0 a 65535
Stop	Define qual é o último ponto de uma faixa que será requisitada. Só pode ser preenchido se a coluna "Qualificador" estiver com o valor "Faixa" selecionado	0	0 a 65535
Count	Define quantos pontos serão requisitados. Só pode ser preenchido se a coluna "Qualificador" estiver com o valor "Contador" selecionado	0	0 a 65535
Polling (ms)	Define o período (em milissegundos) em que ocorre a requisição.	1000	0 a 86400000
Diagnostic Variable	Nome da variável simbólica que receberá o diagnóstico da requisição.	**-**	Nome de uma variável declarada em um programa ou GVL

Tabela 4-75. Configuração das Requisições Customizadas Cliente DNP3

Notas:

Generated Diagnostic Variables: As variáveis simbólicas de diagnósticos podem ser geradas automaticamente através do botão "Gerar Variáveis de Diagnósticos". Ao clicar nesse botão, será criada uma variável do tipo" T_DIAG_DNP_CLIENT_REQUEST_1", na GVL "ReqDiagnostics", para cada uma das requisições.

Qualifier: Identifica o tipo de requisição que será realizada. As opções disponíveis dependem do tipo de Variação do Objeto escolhido. Na Tabela 4-76 está descrito os tipos de Qualificador e na Tabela 4-77 os possíveis valores de qualificador para cada um dos grupos das requisições.

Tipo de Função	Configurações	Descrição
Qualifier	Todos	Todos os pontos do grupo e variação são requisitados
	Faixa	Serão requisitados somente uma faixa de pontos delimitada pelas colunas "Início" e Fim
	Contador	Para requisitar uma quantidade de eventos definida na coluna "Contador", por exemplo, "ler 10 eventos".

Tabela 4-76. Tipos de Qualificador nos Parâmetros do Cliente DNP3

Variação do Objeto	Qualificador Permitido
g01v00 - Entrada digital com variação. g01v01 - Entrada digital em formato compactado. g01v02 - Entrada digital com qualidade.	Todos Faixa

g02v00 - Evento de entrada digital com variação. g02v01 - Evento de entrada digital sem tempo. g02v02 - Evento de entrada digital com tempo absoluto. g02v03 - Evento de entrada digital com tempo relativo.	Todos Contador
g03v00 - Entrada digital dupla com variação. g03v01 - Entrada digital dupla em formato compactado. g03v02 - Entrada digital dupla com qualidade.	Todos Faixa
g04v00 - Evento de entrada digital dupla com variação. g04v01 - Evento de entrada digital dupla sem tempo. g04v02 - Evento de entrada digital dupla com tempo absoluto. g04v03 - Evento de entrada digital dupla com tempo relativo.	Todos Contador
g10v00 - Saída digital com variação. g10v01 - Saída digital em formato compactado. g10v02 - Saída digital com qualidade.	Todos Faixa
g20v00 - Contador com variação. g20v01 - Contador de 32 bits com qualidade. g20v02 - Contador de 16 bits com qualidade. g20v05 - Contador de 32 bits sem qualidade. g20v06 - Contador de 16 bits sem qualidade.	Todos Faixa
g21v00 - Contador congelado com variação. g21v01 - Contador congelado de 32 bits com qualidade. g21v02 - Contador congelado de 16 bits com qualidade. g21v09 - Contador congelado de 32 bits sem qualidade. g21v10 - Contador congelado de 16 bits sem qualidade.	Todos Faixa
g22v00 - Evento do contador com variação. g22v01 - Evento do contador de 32 bits com qualidade. g22v02 - Evento do contador de 16 bits com qualidade. g22v05 - Evento do contador de 32 bits com qualidade e tempo. g22v06 - Evento do contador de 16 bits com qualidade e tempo.	Todos Contador
g23v00 - Evento do contador congelado com variação. g23v01 - Evento do contador congelado de 32 bits com qualidade. g23v02 - Evento do contador congelado de 16 bits com qualidade. g23v05 - Evento do contador congelado de 32 bits com qualidade e tempo. g23v06 - Evento do contador congelado de 16 bits com qualidade e tempo.	Todos Contador
g30v00 - Entrada analógica com variação. g30v01 - Entrada analógica de 32 bits com qualidade. g30v02 - Entrada analógica de 16 bits com qualidade. g30v03 - Entrada analógica de 32 bits sem qualidade. g30v04 - Entrada analógica de 16 bits sem qualidade. g30v05 - Entrada analógica de precisão simples com qualidade.	Todos Faixa
g32v00 - Evento de entrada analógica com variação. g32v01 - Evento de entrada analógica de 32 bits sem tempo. g32v02 - Evento de entrada analógica de 16 bits sem tempo. g32v03 - Evento de entrada analógica de 32 bits com tempo. g32v04 - Evento de entrada analógica de 16 bits com tempo. g32v05 - Evento de entrada analógica de precisão simples sem tempo. g32v07 - Evento de entrada analógica de precisão simples em ponto flutuante com tempo.	Todos Contador
g40v00 - Saída analógica com variação. g40v01 - Saída analógica de 32 bits com qualidade. g40v02 - Saída analógica de 16 bits com qualidade. g40v03 - Saída analógica de precisão simples com qualidade.	Todos Faixa
g60v01 - Classe de dados 0. g60v02 - Classe de dados 1. g60v03 - Classe de dados 2. g60v04 - Classe de dados 3.	Todos Contador

Tabela 4-77. Possíveis Valores de Qualificador para Todos os Grupos de Requisições

Diagnósticos Cliente DNP3

Os diagnósticos e comandos do protocolo DNP3 Client configurado, são armazenados em variáveis do tipo T_DIAG_DNP_CLIENT_ REQUEST_1 as quais estão descritas na Tabela 4-78:

Variável de diagnóstico do tipo T_DIAG_DNP_CLIENT_REQUEST_1.*	Tamanho	Descrição

tCommand	bStop	BOOL	Desabilitar Driver
	bStart	BOOL	Habilitar o Driver
	bDiag_01_Reserved	BOOL	Reservado
	bDiag_02_Reserved	BOOL	Reservado
	bDiag_03_Reserved	BOOL	Reservado
	bDiag_04_Reserved	BOOL	Reservado
	bDiag_05_Reserved	BOOL	Reservado
	bDiag_06_Reserved	BOOL	Reservado
Bits de diagnóstico			
bRunning		BOOL	DNP Cliente está sendo executado
Estatísticas de comunicação			
tStat.*	wRXFrames	WORD	Número de quadros (frames) recebidos
	wTXFrames	WORD	Número de quadros (frames) enviados
	wRequestsSent	WORD	Indica o número de solicitações enviadas. Não contabiliza “solicitações automáticas”
	wCommErrors	WORD	Contador de erros de comunicação incluindo erros na Camada Física, Camada de Enlace e Camada de Transporte.
	dwReserved_0	DWORD	Reservado
	dwReserved_1	DWORD	Reservado
dwReserved_0		DWORD	Reservado
dwReserved_1		DWORD	Reservado
Diagnósticos de Requisição			
eConnection Status	CLOSED	BYTE	Comunicação fechada com o IED
	CONNECTED	BYTE	Comunicação operante com o IED
eRequestStatus	SUCCESS	BYTE	Indicação de que a requisição foi concluída com sucesso
	PROCESSING	BYTE	Indica que uma resposta foi recebida, mas o comando ainda não foi completado. Isto poderia significar que a resposta é parte de uma resposta multi-fragmentada e não tem o bit final setado ou poderia ser um pedido tal como o “select operate” que requer múltiplas solicitações de respostas.
	FAILURE	BYTE	Indica que houve uma falha na transmissão da requisição
	MISMATCH	BYTE	A resposta a um comando “select” ou “operate” indica erros
	STATUS_CODE_CHANGED	BYTE	O status de uma resposta a um comando “select” indica erros.
	IIN_ERROR	BYTE	A resposta de uma requisição possuiu IIN bits indicando erro.
	TIMEOUT	BYTE	Sincronização de tempo é necessária.
	CANCELED	BYTE	Indica que uma segunda requisição foi feita e, portanto, a primeira requisição é cancelada.
tINN.*	ALL_STATIONS	BOOL	Indica que foi recebida uma mensagem de todas as estações.
	CLASS_1_EVENTS	BOOL	Indica que existem eventos de classe 1 na fila do outstation a serem reportados.
	CLASS_2_EVENTS	BOOL	Indica que existem eventos de classe 2 na fila do outstation a serem reportados.
	CLASS_3_EVENTS	BOOL	Indica que existem eventos de classe 3 na fila do outstation a serem reportados.
	NEED_TIME	BOOL	Indica que o IED quer realizar sincronização de tempo usando o DNP3
	LOCAL_CONTROL	BOOL	Um ou mais pontos do outstation estão em modo de controle local.
	DEVICE_TROUBLE	BOOL	Outstation com defeito

	DEVICE_RESTART	BOOL	Outstation foi reiniciado
	NO_FUNC_CODE_SUPPORT	BOOL	O outstation não suporta o código de função
	OBJECT_UNKNOWN	BOOL	Outstation não suporta o comando para os objetos da requisição
	PARAMETER_ERROR	BOOL	Foi detectado um erro de parâmetro
	EVENT_BUFFER_OVERFLOW	BOOL	Existe uma condição de overflow na fila no outstation e pelo menos um evento não confirmado foi perdido.
	ALREADY_EXECUTING	BOOL	A operação solicitada já está em execução
	CONFIG_CORRUPT	BOOL	O outstation possui uma configuração errada
	RESERVED_2	BOOL	Reservado
	RESERVED_1	BOOL	Reservado
dwReserved_0		DWORD	Reservado
dwReserved_1		DWORD	Reservado

Tabela 4-78. Diagnósticos Cliente DNP3

Notas:

Diagnósticos de Requisição: De todos os bits disponibilizados na estrutura tIIN, apenas três deles são efetivamente relacionados à requisição: NO_FUNC_CODE_SUPPORT, OBJECT_UNKNOWN e PARAMETER_ERROR. Todos os outros estão relacionados à Outstation para a qual a requisição está endereçada. Desta forma, caso o usuário deseje avaliar os diagnósticos relacionados a uma Outstation, devem ser consultados os bits de diagnóstico de uma requisição que esteja ativa e que seja periodicamente executada.

Comandos DNP3 para IEDs

Nesta seção, estão listados os comandos de usuário disponíveis para serem enviados via DNP3 para IEDs.

Comando Implementados:

- Código de Função 03: Select
- Código de Função 04: Operate
- Código de Função 05: Direct Operate
- Código de Função 06: Direct Operate – No Acknowledgement
- Código de Função 13: Cold Restart

Devido às características do protocolo, não é suportado o envio de comandos simultâneos, mesmo que destinados a diferentes pontos ou mesmo a diferentes IEDs. Desta forma, ao ser enviado um comando, deve-se aguardar o término da execução do mesmo antes de ser enviado um novo comando. Caso um comando seja disparado enquanto houver algum comando em execução, será retornado o código de erro 0x05 (already active).

Comando para Saída Digital

Parâmetro	Tipo	Descrição
bRequest	BOOL	Quando TRUE, executa o comando
dwVariableAddr	DWORD	Endereço da variável que receberá o comando
eCommand	ENUM (BYTE)	Comando que será enviado: DNP3_COMMAND_TYPE_SELECT (0) DNP3_COMMAND_TYPE_OPERATE (1) DNP3_COMMAND_TYPE_DIRECT_OPERATE (2) DNP3_COMMAND_TYPE_DIRECT_OPERATE_NO_ACK (3)
eOpType	ENUM (BYTE)	DNP3_OP_TYPE_CTRL_NUL (0) DNP3_OP_TYPE_CTRL_PULSE_ON (1) DNP3_OP_TYPE_CTRL_PULSE_OFF (2) DNP3_OP_TYPE_CTRL_LATCH_ON (3) DNP3_OP_TYPE_CTRL_LATCH_OFF (4)

eTripClose	ENUM (BYTE)	DNP3_TRIP_CLOSE_CTRL_NUL (0) DNP3_TRIP_CLOSE_CTRL_PAIRED_CLOSE (64) DNP3_TRIP_CLOSE_CTRL_PAIRED_TRIP (128)
byCount	BYTE	Este é o número de vezes que o client deve executar a operação
udiOnTime	UNSIGNED DOUBLE INT	Esta é a duração, expressa em milissegundos, para que a saída digital permaneça ligada
udiOffTime	UNSIGNED DOUBLE INT	Esta é a duração, expressa em milissegundos, para que a saída digital permaneça desligada
udiCommandTimeOut	UNSIGNED DOUBLE INT	Tempo para o Time Out deste comando

Tabela 4-79. Parâmetros de Entrada Bloco DNP3_BinaryCommand

Parâmetro	Tipo	Descrição
eExecStatus	ENUM (BYTE)	Status da execução, indica se um novo comando pode ser enviado, os valores são os seguintes: DNP3_COMMAND_EXEC_STATUS_DONE (0) DNP3_COMMAND_EXEC_STATUS_RUNNING (1)
eErrorStatus	ENUM (BYTE)	Indica se o último comando foi bem sucedido. DNP3_COMMAND_ERROR_STATUS_NO_ERROR (0) DNP3_COMMAND_ERROR_STATUS_TIMEOUT (1) DNP3_COMMAND_ERROR_STATUS_NO_SELECT (2) DNP3_COMMAND_ERROR_STATUS_FORMAT_ERROR (3) DNP3_COMMAND_ERROR_STATUS_NOT_SUPPORTED (4) DNP3_COMMAND_ERROR_STATUS_ALREADY_ACTIVE (5) DNP3_COMMAND_ERROR_STATUS_HARDWARE_ERROR (6) DNP3_COMMAND_ERROR_STATUS_UNDEFINED (7) DNP3_COMMAND_ERROR_STATUS_INVALID_ADDRESS (8) DNP3_COMMAND_ERROR_STATUS_WRONG_TIMEOUT (9) DNP3_COMMAND_ERROR_STATUS_INVALID_VALUE (10) DNP3_COMMAND_ERROR_STATUS_UNINITIALIZED (11)

Tabela 4-80. Parâmetros de Saída bloco DNP3_BinaryCommand

Notas:

eTripClose: Os comandos de trip/close são exclusivos para os módulos de saída digital que suportem essa funcionalidade. Ex. Módulo HX2320 da Série Hadron Xtorm. Caso o usuário esteja utilizando barramento de expansão misto da Série Hadron Xtorm com a Série Nexto, e utilize um módulo de saída digital que não suporte essa funcionalidade, o código de erro retornado será o "DNP3_COMMAND_ERROR_STATUS_NOT_SUPPORTED".

Comando para Saídas Analógicas

Os comandos para as saídas analógicas são definidos em duas funções: **DNP3_AnalogCommandInt** e **DNP3_AnalogCommandReal.** Na Tabela 4-81 podemos visualizar os parâmetros de entrada e saída para cada um dos blocos de função para as saídas analógicas.

Nome do Bloco de Função: **DNP3_AnalogCommandInt**, usado para números inteiros 16 ou 32 bits. Usados no para o grupo G41V1 e G41V2.

Parâmetro	Tipo	Descrição
bRequest	BOOL	Quando True, executa o comando
dwVariableAddr	DWORD	Endereço da variável de comando para enviar
eCommand	ENUM (BYTE)	Comando que será enviado: DNP3_COMMAND_TYPE_SELECT (0) DNP3_COMMAND_TYPE_OPERATE (1) DNP3_COMMAND_TYPE_DIRECT_OPERATE (2) DNP3_COMMAND_TYPE_ DIRECT_OPERATE_NO_ACK (3)
dValue	DOUBLE INT	Valor analógico que será escrito: INT ou DINT
udiCommandTimeOut	UNSIGNED DOUBLE INT	Tempo para o Time Out deste comando

Tabela 4-81. Parâmetros de Entrada bloco DNP3_AnalogCommandInt

Parâmetro	Tipo	Descrição
eExecStatus	ENUM (BYTE)	Status da execução, indica se um novo comando pode ser enviado, os valores são os seguintes: DNP3_COMMAND_EXEC_STATUS_DONE (0) DNP3_COMMAND_EXEC_STATUS_RUNNING (1)
eErrorStatus	ENUM (BYTE)	Indica se o último comando foi bem sucedido. DNP3_COMMAND_ERROR_STATUS_NO_ERROR (0) DNP3_COMMAND_ERROR_STATUS_TIMEOUT (1) DNP3_COMMAND_ERROR_STATUS_NO_SELECT (2) DNP3_COMMAND_ERROR_STATUS_FORMAT_ERROR (3) DNP3_COMMAND_ERROR_STATUS_NOT_SUPPORTED (4) DNP3_COMMAND_ERROR_STATUS_ALREADY_ACTIVE (5) DNP3_COMMAND_ERROR_STATUS_HARDWARE_ERROR (6) DNP3_COMMAND_ERROR_STATUS_UNDEFINED (7) DNP3_COMMAND_ERROR_STATUS_INVALID_ADDRESS (8) DNP3_COMMAND_ERROR_STATUS_WRONG_TIMEOUT (9) DNP3_COMMAND_ERROR_STATUS_INVALID_VALUE (10) DNP3_COMMAND_ERROR_STATUS_UNINITIALIZED (11)

Tabela 4-82. Parâmetros de Saída bloco DNP3_AnalogCommandInt

Nome do Bloco de Função: **DNP3_AnalogCommandReal**, usado para números do tipo real 32 bits. Usado no para o grupo G41V3.

Parâmetro	Tipo	Descrição
bRequest	BOOL	Quando True, executa o comando
dwVariableAddr	DWORD	Endereço da variável de comando para enviar
eCommand	ENUM (BYTE)	Comando que será enviado: DNP3_COMMAND_TYPE_SELECT (1) DNP3_COMMAND_TYPE_OPERATE (2) DNP3_COMMAND_TYPE_DIRECT_OPERATE (3) DNP3_COMMAND_TYPE_DIRECT_OPERATE_NO_ACK (4)
rValue	REAL	Valor analógico que será escrito: REAL
udiCommandTimeOut	UNSIGNED DOUBLE INT	Tempo para o Time Out deste comando

Tabela 4-83. Parâmetros de Entrada do bloco DNP3_AnalogCommandReal

Parâmetro	Tipo	Descrição
eExecStatus	ENUM (BYTE)	Status da execução, indica se um novo comando pode ser enviado, os valores são os seguintes: DNP3_COMMAND_EXEC_STATUS_DONE (0) DNP3_COMMAND_EXEC_STATUS_RUNNING (1)
eErrorStatus	ENUM (BYTE)	Indica se o último comando foi bem sucedido. DNP3_COMMAND_ERROR_STATUS_NO_ERROR (0) DNP3_COMMAND_ERROR_STATUS_TIMEOUT (1) DNP3_COMMAND_ERROR_STATUS_NO_SELECT (2) DNP3_COMMAND_ERROR_STATUS_FORMAT_ERROR (3) DNP3_COMMAND_ERROR_STATUS_NOT_SUPPORTED (4) DNP3_COMMAND_ERROR_STATUS_ALREADY_ACTIVE (5) DNP3_COMMAND_ERROR_STATUS_HARDWARE_ERROR (6) DNP3_COMMAND_ERROR_STATUS_UNDEFINED (7) DNP3_COMMAND_ERROR_STATUS_INVALID_ADDRESS (8) DNP3_COMMAND_ERROR_STATUS_WRONG_TIMEOUT (9) DNP3_COMMAND_ERROR_STATUS_INVALID_VALUE (10) DNP3_COMMAND_ERROR_STATUS_UNINITIALIZED (11)

Tabela 4-84. Parâmetros de Saída do bloco DNP3_AnalogCommandReal

Comando Cold Restart

Nome do Bloco de Função: **DNP3_ColdCommand**

Parâmetro	Tipo	Descrição
bRequest	BOOL	Quando True, executa o comando

dwVariableAddr	DWORD	Endereço da variável de comando para enviar
udiCommandTimeOut	UNSIGNED DOUBLE INT	Tempo para o Time Out deste comando

Tabela 4-85. Parâmetros de Entrada do bloco DNP3_ColdCommand

Parâmetro	Tipo	Descrição
eExecStatus	ENUM (BYTE)	Status da execução, indica se um novo comando pode ser enviado, os valores são os seguintes: DNP3_COMMAND_EXEC_STATUS_DONE (0) DNP3_COMMAND_EXEC_STATUS_RUNNING (1)
eErrorStatus	ENUM (BYTE)	Indica se o último comando foi bem sucedido. DNP3_COMMAND_ERROR_STATUS_NO_ERROR (0) DNP3_COMMAND_ERROR_STATUS_TIMEOUT (1) DNP3_COMMAND_ERROR_STATUS_NO_SELECT (2) DNP3_COMMAND_ERROR_STATUS_FORMAT_ERROR (3) DNP3_COMMAND_ERROR_STATUS_NOT_SUPPORTED (4) DNP3_COMMAND_ERROR_STATUS_ALREADY_ACTIVE (5) DNP3_COMMAND_ERROR_STATUS_HARDWARE_ERROR (6) DNP3_COMMAND_ERROR_STATUS_UNDEFINED (7) DNP3_COMMAND_ERROR_STATUS_INVALID_ADDRESS (8) DNP3_COMMAND_ERROR_STATUS_WRONG_TIMEOUT (9) DNP3_COMMAND_ERROR_STATUS_INVALID_VALUE (10) DNP3_COMMAND_ERROR_STATUS_UNINITIALIZED (11)

Tabela 4-86. Parâmetros de Saída do bloco DNP3_ColdCommand

Exemplo de Aplicação:

Abaixo um exemplo de aplicação utilizando a linguagem ST, na qual o usuário envia o comando Select Before Operate para um IED via DNP3.

```
PROGRAM UserPrg
VAR
    SendAnalogOutReal : DNP3_AnalogCommandReal;
END_VAR

SendAnalogOutReal.dwVariableAddr := ADR( GVLData.SingleAnalogOutReal);
SendAnalogOutReal();
IF( SendAnalogOutReal.eExecStatus = DNP3_COMMAND_EXEC_STATUS_DONE) THEN
    IF SendAnalogOutReal.bRequest = TRUE
    AND   SendAnalogOutReal.eCommand = DNP3_COMMAND_TYPE_SELECT
    AND SendAnalogOutReal.eErrorStatus = DNP3_COMMAND_ERROR_STATUS_NO_ERROR
    THEN
          SendAnalogOutReal.eCommand := DNP3_COMMAND_TYPE_OPERATE;
    ELSE
          SendAnalogOutReal.bRequest := FALSE;
    END_IF
    END_IF
```

Códigos de Status para Comandos DNP

Muitos dos comandos DNP3 podem retornar códigos de status. Os códigos de status aparecem nas respostas para os comandos de saída, associados com blocos de controle, blocos de saída analógica e eventos de comando de saída. Na Tabela 4-87 podemos visualizar uma lista de códigos de estados válidos:

Número do Código	Nome	Descrição
0	SUCCESS	Pedido aceito, iniciado ou na fila.
1	TIMEOUT	Pedido não aceito porque a mensagem foi recebida após o time out do comando select.
2	NO_SELECT	Pedido não aceito porque o comando select

		não foi enviado previamente.
3	FORMAT_ERROR	Pedido não aceito porque havia erros de formatação do pedido de controle (select, operate ou direct operate).
4	NOT_SUPPORTED	Pedido não aceito, porque uma operação de controle não é suportada por este ponto.
5	ALREADY_ACTIVE	Pedido não aceito porque a fila de controle já está cheia ou o ponto já está ativo.
6	HARDWARE_ERROR	Pedido não aceito, por causa de problemas de hardware.
7	LOCAL	Pedido não aceito porque a “chave” Local/Remoto está em posição Local.
8	TOO_MANY_OBJS	Pedido não aceito porque muitos objetos estão presentes no mesmo pedido.
9	NOT_AUTHORIZED	Código não implementado.
10	AUTOMATION_INHIBIT	Pedido não aceito porque foi impedido ou inibido por um processo de automação local.
11	PROCESSING_LIMITED	Pedido não aceito porque o dispositivo não pode processar mais nenhuma atividade, além daquelas em andamento.
12	OUT_OF_RANGE	Pedido não aceito porque o valor está fora da faixa aceitável permitido para este ponto.
65	REMOTE_TIMEOUT	Time-out no IED remoto
13 a 125	RESERVED	Reservados para uso futuro.
126	NON_PARTICIPATING	Código não implementado.
127	UNDEFINED	Pedido não aceite por causa de algum outro motivo indefinido.

Tabela 4-87. Códigos de status para comandos DNP

DNP3 Ethernet Servidor

Este protocolo está disponível para a UCP da Série Hadron Xtorm nos seus canais Ethernet. Ao selecionar esta opção no MasterTool Xtorm, a UCP passa a ser servidor da comunicação DNP3, permitindo a conexão com até quatro dispositivos clientes DNP3. Para cada cliente, o driver possui uma fila de eventos exclusiva com as seguintes características:

- Tamanho: 4.500 eventos
- Política de estouro: mantém o mais recente

Para configurar este protocolo, é necessário executar os seguintes passos:

- Adicionar a instância do protocolo Servidor DNP3 a um dos canais Ethernet disponíveis (NET 1 ao NET 6)
- Para realizar esse procedimento, consultar o capítulo Inserindo uma Instância de Protocolo
- Configurar a interface Ethernet. Para realizar esse procedimento, consultar o capítulo Configuração das Interfaces Ethernet
- Configurar os parâmetros gerais do protocolo Servidor DNP3, com o modo de conexão Porta ou IP, e o número da porta TCP quando o modo de conexão selecionado for o modo IP
- Adicionar e configurar dispositivos, definindo devidos parâmetros
- Adicionar e configurar os mapeamentos DNP3, especificando o nome da variável, variação do grupo estático, variação do grupo de eventos, índice, tamanho, faixa, classe do evento, banda morta, e tipo da banda morta
- Configurar os parâmetros da camada de enlace, especificando os endereçamentos e as mensagens de confirmação
- Configurar os parâmetros da camada de aplicação, especificando a fragmentação de respostas, mensagens de confirmação, polling de integridade, sincronismo, comandos e fila de eventos
- Configurar os parâmetros das mensagens não solicitadas, especificando os parâmetros de controle e as condições de disparo

As descrições de cada configuração estão relacionadas a seguir, neste capítulo.

Configuração dos Mapeamentos Servidor DNP3

As configurações dos Parâmetros Gerais de um Servidor DNP3 conforme a Figura 4-29 seguem os parâmetros da Tabela 4-88 apresentadas a seguir:

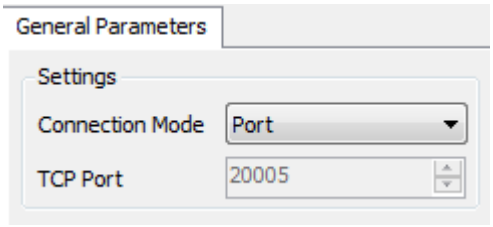


Figura 4-29. Tela de Parâmetros Gerais do Servidor DNP3

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Connection Mode	Configura o modo de conexão com os módulos Cliente Conectados. Os modos disponíveis para conexão são "IP" ou "Porta".	Port	Port IP
TCP Port	Define qual o número da porta TCP da UTR será utilizada para a comunicação com os módulos Cliente Conectados, caso no campo "Modo de Conexão" esteja selecionado "IP".	20005	1 a 65535

Tabela 4-88. Configuração dos Parâmetros Gerais do Servidor DNP3

A configuração das relações Servidor DNP3, visualizadas na Figura 4-30, segue os parâmetros descritos na Tabela 4-89:

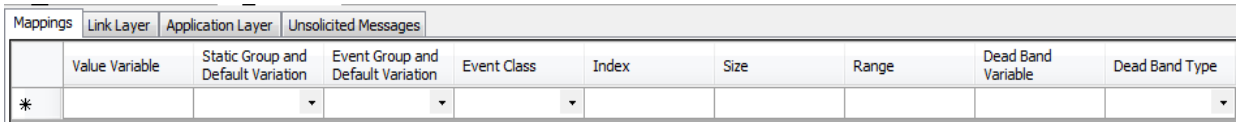


Figura 4-30. Tela de Mapeamentos de Dados Servidor DNP3

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Value Variable	Nome da variável simbólica	-	Nome de uma variável declarada em um programa ou GVL
Static Group and Default Variation	Configuração da variação dos dados estáticos	-	g10v1 - Saída digital em formato compactado g10v2 - Saída digital com qualidade g1v1 - Entrada digital em formato compactado g1v2 - Entrada digital com qualidade g20v1 – Contador de 32 bits com qualidade g20v2 – Contador de 16 bits com qualidade g20v5 – Contador de 32 bits sem qualidade g20v6 – Contador de 16 bits sem qualidade g21v1 – Contador congelado de 32 bits com qualidade g21v10 – Contador congelado de 16 bits sem qualidade g21v2 – Contador congelado de 16 bits com qualidade g21v9 – Contador congelado de 32 bits sem qualidade g30v1 – Entrada analógica de 32 bits com qualidade g30v2 – Entrada analógica de 16 bits com qualidade g30v3 – Entrada analógica de 32 bits sem qualidade

			g30v4 – Entrada analógica de 16 bits sem qualidade g30v5 – Entrada analógica de precisão simples com qualidade g3v1 – Entrada digital dupla em formato compactado g3v2 – Entrada digital dupla com qualidade g40v1 – Saída analógica de 32 bits com qualidade g40v2 – Saída analógica de 16 bits com qualidade g40v3 – Saída analógica de precisão simples com qualidade
Event Group and Default Variation	Configuração da variação para os eventos	-	g22v1 – Evento do contador de 32 bits com qualidade g22v2 – Evento do contador de 16 bits com qualidade g22v5 – Evento do contador de 32 bits com qualidade e tempo g22v6 – Evento do contador de 16 bits com qualidade e tempo g23v1 – Evento do contador congelado de 32 bits com qualidade g23v2 – Evento do contador congelado de 16 bits com qualidade g23v5 – Evento do contador congelado de 32 bits com qualidade e tempo g23v6 – Evento do contador congelado de 16 bits com qualidade e tempo g2v1 – Evento de entrada digital sem tempo g2v2 – Evento de entrada digital com tempo absoluto g2v3 – Evento de entrada digital com tempo relativo g32v1 – Evento de entrada analógica de 32 bits sem tempo g32v2 – Evento de entrada analógica de 16 bits sem tempo g32v3 – Evento de entrada analógica de 32 bits com tempo g32v4 – Evento de entrada analógica de 16 bits com tempo g32v5 – Evento de entrada analógica de precisão simples sem tempo g4v1 – Evento de entrada digital dupla sem tempo g4v2 – Evento de entrada digital dupla com tempo absoluto g4v3 – Evento de entrada digital com tempo relativo Nenhum
Event Class	Configuração da Classe do Evento	-	Classe 0 Classe 1 Classe 2 Classe 3
Index	Índice do primeiro ponto de mapeamento DNP3	-	0 a 65535
Size	Tamanho do dado DNP3	-	1 a 86400000
Range	Faixa de endereços do dado configurado	-	-
Dead Band Variable	Nome da variável simbólica que vai conter o dado da banda morta	-	Nome de uma variável declarada em um programa ou GVL
Dead Band Type	Define o tipo de Banda Morta a ser utilizada no mapeamento	Desabilitada	Absoluta Desabilitada Integrada

Tabela 4-89. Configuração dos Mapeamentos Servidor DNP3

Notas:

Value Variable: Nome da variável simbólica a ser mapeada, Quando um comando de leitura é enviado, o retorno enviado na resposta será armazenado nesta variável. Quando for um comando de escrita, o valor escrito será armazenado nesta variável. A variável pode ser simples, array ou elemento de array e pode estar em estruturas.

Static Group and Default Variation: O campo é utilizado para configurar a variação dos dados estáticos que serão retornados caso o cliente DNP3 não especifique uma variação na requisição dos dados, ou no caso de mensagens não solicitadas.

Event Group and Default Variation: O campo é utilizado para configurar a variação para os eventos que serão retornados caso o cliente DNP3 não especifique uma variação na requisição dos dados, ou no caso de mensagens não solicitadas.

Event Class: O campo é utilizado para configurar em qual classe os eventos deste mapeamento serão armazenados. Sendo que para pontos configurados como classe 0, não serão reportados eventos. Lembrando que para os grupos 10 e 40, o valor deste campo é fixo em zero, visto que tratam de saídas digital e analógicas respectivamente.

Index: O campo é utilizado para configurar o índice do primeiro ponto do mapeamento no protocolo DNP3.

Size: O valor de Tamanho especifica a quantidade máxima de dados que uma relação DNP3 poderá acessar, a partir do endereço inicial. Sendo assim, para ler uma faixa de endereços contínua, é necessário que todos os endereços estejam declarados em uma única relação. Este campo varia de acordo com o tipo de dado DNP3 configurado.

Range: Este campo mostra ao usuário a faixa de endereços de memória utilizada pela relação DNP3.

Dead Band Variable: Este campo é utilizado para especificar uma variável que vai conter os dados da banda morta da relação DNP3.

Dead Band Type: Os tipos de configuração da Banda Morta disponíveis são os seguintes:

Tipo de Função	Configurações	Descrição
Tipo da Banda Morta	Desabilitada	Nesta opção, qualquer modificação de valor num ponto do grupo, por menor que seja, gera evento para este ponto.
	Absoluta	Nesta opção, se o módulo da variação do valor de um ponto do grupo for maior do que o valor configurado pela variável no campo “Banda Morta”, um evento para este ponto é gerado.
	Integrada	Nesta opção, se o módulo da integral da variação do valor de um ponto do grupo for maior do que o valor configurado pela variável no campo “Banda Morta”, um evento para este ponto é gerado.

Tabela 4-90. Tipos de Banda Morta nos parâmetros do Servidor DNP3

Configuração da Camada de Enlace Servidor DNP3

A configuração dos parâmetros da camada de enlace do Servidor DNP3, visualizadas na Figura 4-31, segue os parâmetros descritos na Tabela 4-91:

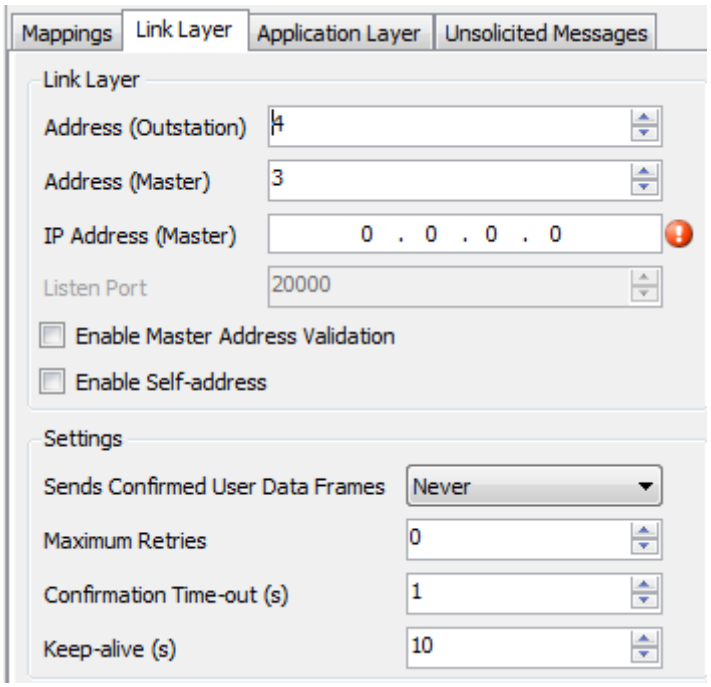


Figura 4-31. Tela de Configurações da Camada de Enlace Servidor DNP3

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Address (Outstation)	Endereço DNP3 deste servidor	4	0 a 65519
Address (Master)	Endereço DNP3 do cliente conectado.	3	0 a 65519
IP Address (Master)	IP do cliente conectado, utilizado quando a conexão do cliente é por IP	0.0.0.0	1.0.0.1 a 223.255.255.254
Listen Port	Endereço da porta listen para conexão do cliente. Utilizado quando a conexão do cliente não é por IP	2000	1 a 65535
Enable Master Address Validation	Opção para Habilitar/Desabilitar a validação do endereço de Origem	Disabled	Disabled Enabled
Enable Self-address	Opção para Habilitar/Desabilitar o Auto Endereçamento	Disabled	Disabled Enabled
Sends Confirmed User Data Frames	Seleciona o modo de confirmações para as mensagens enviadas para o cliente	Never	Never Always Sometimes
Maximum Retries	Número Máximo de Tentativas de retransmissão de mensagem	0	0 a 10
Confirmation Time-out (s)	Tempo de espera para enviar uma nova tentativa	1	1 a 86400
Keep Alive (s)	Tempo de espera para enviar mensagens de teste de link após receber a mensagem DNP	1	1 a 86400

Tabela 4-91. Configuração da Camada de Enlace Servidor DNP3

Notas:

Address (Outstation): O campo é utilizado para configurar o endereço DNP3 deste servidor.

Address (Master): O campo é utilizado para configurar o endereço DNP3 do cliente.

IP Address (Master): O campo é utilizado para configurar o endereço IP do cliente conectado, utilizado quando a conexão do cliente é por IP

Listen Port: Endereço da porta listen para conexão do cliente. Utilizado quando a conexão do cliente não é por endereçamento IP.

Enable Master Address Validation: Se essa opção estiver marcada, o servidor DNP3 só irá aceitar mensagens cujos endereços de origem e destinos coincidem com os valores configurados respectivamente nos campos “Endereço de Origem” e “Endereço de Destino”. Ou seja o campo “Endereço de Origem” da mensagem deve coincidir com o valor do campo “Endereço de Destino”. configurado para o Cliente. E o campo “Endereço de Destino” da mensagem deve coincidir com o valor do campo “Endereço de Origem” configurado pelo cliente

Enable Self-address: Habilita a recepção de mensagens com o endereço 65532 do cliente DNP3

Sends Confirmed User Data Frames: Os modos de confirmação da camada de enlace disponíveis são os seguintes:

Tipo de Função	Configurações	Descrição
Modo de Confirmação da Camada de Enlace	Nunca	Nunca solicitará confirmação das mensagens
	Sempre	Sempre solicitará confirmação das mensagens
	Ás vezes	Solicitará confirmação somente para mensagens multifragmentadas

Tabela 4-92. Tipos de Modo de Confirmação da Camada de Enlace nos parâmetros do Servidor DNP3

Maximum Retries: Número de vezes que o Servidor DNP irá retransmitir a mensagem caso ele não receba a confirmação do cliente

Confirmation Time-Out (s): Tempo de espera para enviar uma nova tentativa

Keep Alive (s): Intervalo de tempo para envio de mensagens de teste de link ("TCP keep-alive" ou "serial link status") após última mensagem DNP recebida. Caso não seja recebida uma resposta para a mensagem de teste de link, a conexão atual será encerrada. Este mecanismo de teste deve ser corretamente configurado, pois é através dele que problemas de conexão são identificados.

Configuração da Camada de Aplicação Servidor DNP3

A configuração das relações Servidor DNP3, visualizadas na Figura 4-32, segue os parâmetros descritos na Tabela 4-93:

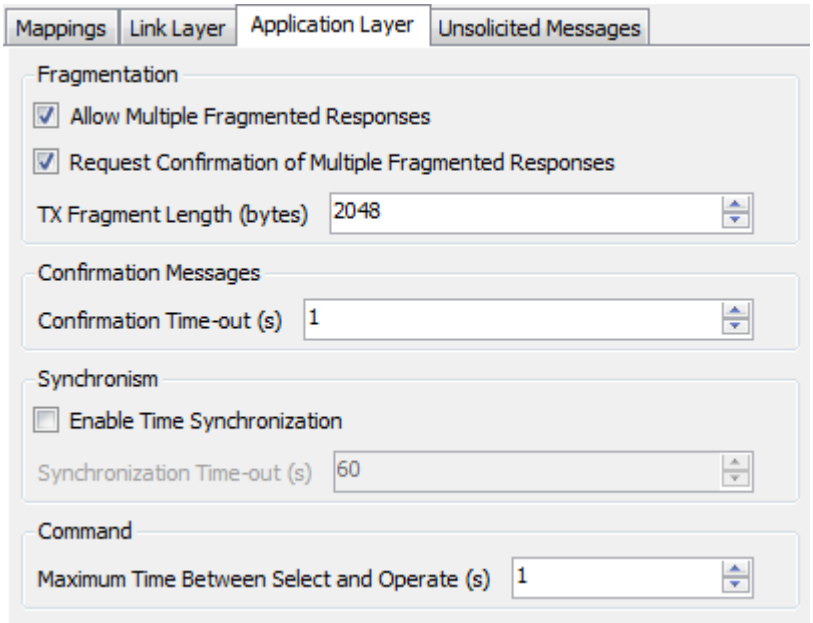


Figura 4-32. Tela de Configurações da Camada de Aplicação Servidor DNP3

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Allow Multiple Fragmented	Opção para Habilitar/Desabilitar a Transmissão de Respostas Múltiplos	Enabled	Disabled Enabled

Responses	Fragmentos		
Request Confirmation of Multiple Fragmented Responses	Opção para Habilitar/Desabilitar Requisição de Confirmação de Resposta com Múltiplos Fragmentos	Enabled	Disabled Enabled
TX Fragmented Length (bytes)	Tamanho máximo dos fragmentos	2048	249 a 2948
Confirmation Time-out (s)	Determina qual é o tempo máximo de espera para as mensagens de confirmação.	1	1 a 86400
Enable Time Synchronization	Opção para Habilitar/Desabilitar a solicitação de sincronismo de tempo	Disabled	Disabled Enabled
Synchronization Time-out (s)	Define o tempo que o relógio local continuará válido após ter recebido uma sincronização de tempo	60	1 a 86400
Maximum Time Between Selected and Operate (s)	Tempo máximo que o comando select permanecerá ativo aguardando o comando Operate	1	1 a 86400

Tabela 4-93. Configuração da Camada de Aplicação Servidor DNP3

Notas:

Allow Multiple Fragmented Responses: Se essa opção estiver marcada, o servidor DNP3 irá permitir a divisão de uma mensagem de aplicação em mensagens menores e enviá-las uma a uma.

Request Confirmation of Multiple Fragmented Responses: Se essa opção estiver marcada, o servidor DNP3 irá requisitar ao cliente uma confirmação que recebeu a mensagem com o fragmento da mensagem de aplicação.

TX Fragmented Length (bytes): Máximos de tamanhos de fragmentos transmitidos aplicação.

Confirmation Time-out (s): O time-out de mensagens de confirmação é o tempo que o servidor aguardará por uma confirmação de recebimento da mensagem pelo cliente.

Enable Time Synchronization: Se essa opção estiver marcada, o servidor DNP3 estará habilitado a solicitar o sincronismo de tempo do cliente. Esta solicitação é feita na primeira mensagem transmitida ao cliente depois que estourar o tempo configurado no campo "Time-Out de Sincronismo".
Synchronization Time-out (s): Este campo é utilizado para configurar o intervalo de tempo para as solicitações de sincronização de tempo.

Maximum Time Between Selected and Operate (s): Período de tempo no qual o comando de seleção permanecerá ativo (inicia a contagem a partir da confirmação da recepção do comando de seleção).

ATENÇÃO:
Quando a opção de sincronismo de tempo for marcada em mais de um outstation (parâmetro "Habilitar Sincronismo") os tempos recebidos nos diferentes outstations serão sobrescritos no relógio do sistema em um curto espaço de tempo, podendo causar comportamentos indesejáveis devido a atrasos no tempo de propagação das mensagens e carga do sistema.

Configuração das Mensagens Não Solicitadas Servidor DNP3

A configuração das relações Servidor DNP3, visualizadas na Figura 4-33, segue os parâmetros descritos na Tabela 4-94:

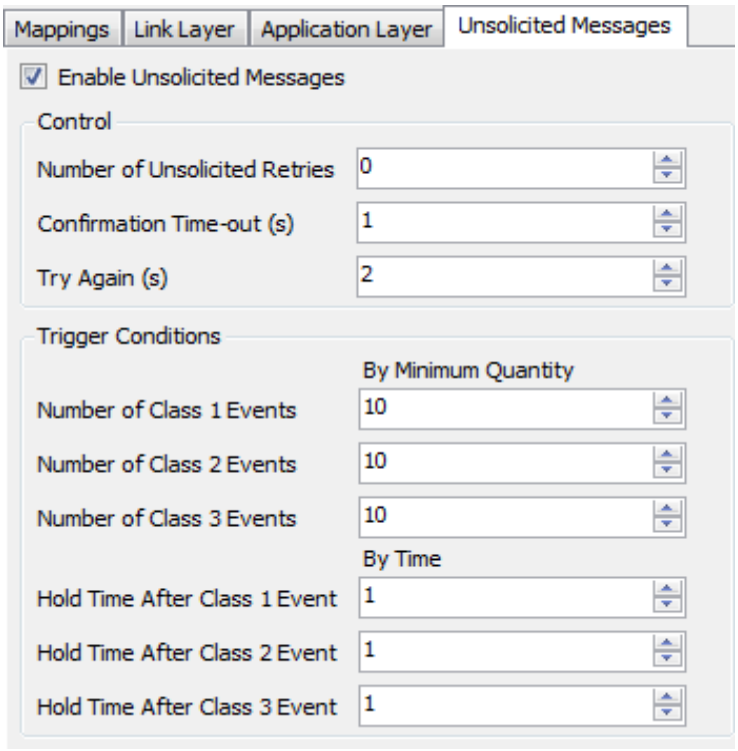


Figura 4-33. Tela de Mensagens Não Solicitadas Servidor DNP3

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Enable Unsolicited Messages	Opção para Habilitar/Desabilitar as mensagens não solicitadas	Enabled	Disabled Enabled
Number of Unsolicited Retries	Número de tentativas para transmitir a mensagem	0	0 a 65535
Confirmation Time-out (s)	Intervalo a ser aguardado depois de um time out.	1	1 a 86400
Try Again (s)	Intervalo a ser aguardado após atingir o número máximo de tentativas.	2	1 a 86400
Number of Class 1 Events	Quantidade mínimo de eventos na Classe 1 para disparar uma mensagem não solicitada	10	1 a 255
Number of Class 2 Events	Quantidade mínimo de eventos na Classe 2 para disparar uma mensagem não solicitada	10	1 a 255
Number of Class 3 Events	Quantidade mínimo de eventos na Classe 3 para disparar uma mensagem não solicitada	10	1 a 255
Hold Time After Class 1 Event	Tempo máximo que um evento pode ficar armazenado na classe 1 sem ser enviado por uma mensagem não solicitada	1	1 a 86400
Hold Time After Class 2 Event	Tempo máximo que um evento pode ficar armazenado na classe 2 sem ser enviado por uma mensagem não solicitada	1	1 a 86400
Hold Time After Class 3 Event	Tempo máximo que um evento pode ficar armazenado na classe 3 sem ser enviado por uma mensagem não solicitada	1	1 a 86400

Tabela 4-94. Configuração das Mensagens Não Solicitadas Servidor DNP3

Notas:

Enable Unsolicited Messages: Se essa opção estiver marcada, o servidor DNP3 irá permitir Mensagens não solicitadas

Number of Unsolicited Retries: Número de vezes que o servidor DNP3 irá tentar transmitir a mensagem.

Confirmation Time-out(s): Especifica o tempo a ser aguardado depois de ocorrer um time-out para a mensagem não solicitada que foi transmitida. Transcorrido esse intervalo a mensagem será retransmitida. Logo o tempo de retransmissão, será dado pelos parâmetros : "**Time-out de Mensagens de Confirmação" + "Time-Out".**

Try Again (s): Após o número máximo de tentativas para transmitir a mensagem ser atingido (ver parâmetro "Number of Unsolicited Retries"), as retransmissões restantes serão baseadas nesse parâmetro de tempo. Sendo assim, o tempo de retransmissão passará a ser dado pelos parâmetros **"Time-out de Mensagens de Confirmação" + "Tentar Novamente"**.

Number of Class 1 Events: Quantidade mínimo de eventos na classe 1 para disparar a mensagem não solicitada.

Number of Class 2 Events: Quantidade mínimo de eventos na classe 2 para disparar a mensagem não solicitada.

Number of Class 3 Events: Quantidade mínimo de eventos na classe 3 para disparar a mensagem não solicitada.

Hold Time After Class 1 Event: Tempo máximo que um evento da classe 1 pode ficar armazenado sem ser enviado por uma mensagem não solicitada.

Hold Time After Class 2 Event: Tempo máximo que um evento da classe 2 pode ficar armazenado sem ser enviado por uma mensagem não solicitada.

Hold Time After Class 3 Event: Tempo máximo que um evento da classe 3 pode ficar armazenado sem ser enviado por uma mensagem não solicitada.

Diagnósticos Servidor DNP3

Os diagnósticos e comandos do protocolo Servidor DNP3 configurado, são armazenados em variáveis do tipo T_DIAG_DNP_SERVER_1 as quais estão descritas na Tabela 4-95:

Variável de diagnóstico do tipo T_DIAG_DNP_SERVER_1.*			Tamanho	Descrição
Bits de comando, reiniciados automaticamente:				
tCommand		bStop	BOOL	Desabilitar Driver
		bStart	BOOL	Habilitar o Driver
		bDiag_01_Reserved	BOOL	Reservado
		bDiag_02_Reserved	BOOL	Reservado
		bDiag_03_Reserved	BOOL	Reservado
		bDiag_04_Reserved	BOOL	Reservado
		bDiag_05_Reserved	BOOL	Reservado
		bDiag_06_Reserved	BOOL	Reservado
Diagnósticos				
tClient_X	**bRunning**		BOOL	DNP Server está sendo executado
	eConnectionStatus	CLOSED	ENUM (BYTE)	Canal de comunicação fechado. Servidor não irá aceitar solicitação de conexão. Valor do ENUM (0)
		LISTENING		Servidor está escutando a porta configurada, e não existem clientes

				conectados. Valor do ENUM (1)
		CONNECTED		Cliente conectado. Valor do ENUM (2)
	tQueueDiags	bOverflow	BOOL	Fila do cliente está com overflow
		wSIZE	WORD	Tamanho da fila configurada
		wUSAGE	WORD	Número de eventos na fila
		dwReserved_0	DWORD	Reservado
		dwReserved_1	DWORD	Reservado
	tStats	wRXFrames	WORD	Número de quadros (frames) recebidos
		wTXFrames	WORD	Número de quadros (frames) enviados
		wCommErrors	WORD	Contador de erros de comunicação incluindo erros na Camada Física, Camada de Enlace e Camada de Transporte.
		dwReserved_0	DWORD	Reservado
		dwReserved_1	DWORD	Reservado

Tabela 4-95. Diagnósticos DNP3 Server

Comando Cold Restart

O comando de Cold Restart força a reinicialização da remota. Os eventos da fila interna não são perdidos, ou seja, os valores das variáveis do tipo retentivas e persistentes não são alterados. Porém os eventos, que por motivo de overflow, estejam apenas nas filas dos drivers, serão perdidos. Uma vez que a remota é reinicializada, todas as configurações realizadas em tempo de execução, específicas de cada protocolo, serão perdidas, e as configurações vigentes serão aquelas originais do projeto, realizadas com o auxílio do software MasterTool.

Comandos para Pontos de Saída e Contadores

Os comandos recebidos pelo driver Servidor DNP3 podem ter os seguintes destinos:

- Pontos Internos
- Módulo de saída (por ex. HX2320)
- Driver Cliente DNP3
- Interceptação (bloco de função CommandReceiver)

Comandos para congelamento de contadores para uma faixa de pontos são apenas suportados caso todos os pontos contidos na faixa sejam Pontos Internos. Caso um ou mais destes pontos não seja interno (redirecionado para um IED conectado a um driver DNP3 cliente por exemplo), o comando não será aceito. Nestes casos, o comando de congelamento deve ser executado individualmente para cada ponto.

Pontos Internos

Pontos internos são considerados aqueles pontos representados por variáveis que não estão associadas a nenhum dos outros destinos descritos acima. Para este tipo de ponto, são suportados apenas comandos para pontos simples. Caso o Servidor DNP3 receba um comando trip/close para um ponto interno, será retornado o código de erro 0x04 (not supported).

Módulo de Saída

Caso a variável mapeada para o ponto de saída DNP3 esteja associada a um cartão de saída (por ex. HX2320), o comando será redirecionado e executado pelo próprio cartão. No caso de comandos trip/close, este redirecionamento funcionará apenas se a variável for do tipo DBP. Caso seja utilizada uma variável do tipo BOOL, o Servidor DNP3 retornará o código de erro 0x03 (wrong format), assim como quando um comando Latched é direcionado a uma variável DBP mapeada no cartão de saída.

Driver Cliente DNP3

Caso a variável esteja associada a um ponto de um IED em um driver DNP3 Client, o comando será redirecionado e enviado para ser executado pelo próprio IED e o valor da variável armazenada na memória da UCP não é atualizado. No caso de comandos trip/close, tipicamente a variável deve ser do tipo DBP, contudo o redirecionamento funcionará da mesma forma mesmo que seja utilizada uma variável do tipo BOOL.

Deve-se considerar que, devido às características do protocolo, o driver Cliente DNP3 é capaz de executar somente um comando por vez. Isto é, enquanto um comando para um determinado ponto não for finalizado (resposta ou time-out), não é possível disparar um novo comando, mesmo que seja para um ponto diferente. Desta forma, caso um comando seja redirecionado do Servidor DNP3 para um IED associado a um driver Cliente DNP3 e este esteja ocupado executando um comando, será retornado o código de erro 0x05 (already active).

Interceptação de Comandos

A interceptação de comandos recebidos pelo driver DNP3 Servidor é realizada através do bloco de função CommandReceive conforme descrito na seção Interceptação de Comandos Oriundos do Centro de Controle. Este recurso permite a interceptação de comandos de seleção e operação (isto é, funções 3, 4, 5 e 6 do protocolo DNP3).

Abaixo um exemplo de aplicação utilizando a linguagem ST mostrando a interceptação de um comando para um ponto DNP3 mapeado para a variável “SinglePoint”.

```
PROGRAM UserPrg
VAR
    CommandReceiver : LibRtuStandard.CommandReceiver;
END_VAR

CommandReceiver.dwVariableAddr := ADR( GVLData.SinglePoint);
CommandReceiver();
IF( CommandReceiver.bCommandAvailable = TRUE) THEN
( *
Processa o comando
A interceptação é encerrada quando retornado o status em
CommandReceiver.eCommandResult
e sinalizado o fim da interceptação em CommandReceiver.bDone
*)
END_IF
```

Desempenho da Comunicação DNP3

DNP3 Servidor

O driver Servidor DNP3 é executado pela UCP da mesma forma como os outros drivers de comunicação Servidores, isto é, em segundo plano, com uma prioridade abaixo da aplicação de usuário e de forma cíclica. A tarefa deste driver especificamente executa a cada 50ms, e basta 1 ciclo de execução do driver para processar e responder requisições. Contudo, por se tratar de uma tarefa de baixa prioridade, não é garantido que consiga executar com esta periodicidade pois depende do percentual de UCP livre (diferença entre o intervalo da MainTask e tempo que a aplicação do usuário leva para ser executada) e também da concorrência com tarefas de outros protocolos configurados na UCP.

Para auxiliar na compreensão do desempenho do driver DNP3 Servidor, a seguir é apresentado o resultado de testes realizados com um simulador Cliente DNP3 conectado a uma HX3040 com um driver DNP3 Servidor. A base de dados configurada era composta de 4000 pontos digitais e 500 pontos analógicos (todos com qualidade e estampa de tempo), e o tempo de execução da MainTask estava em 70ms (interval de 100ms).

- Tempo para completar um polling de integridade: 1segundo
- Tempo para transferir 4.000 eventos digitais + 500 eventos analógicos: 7 segundos

DNP3 Cliente

O driver Cliente DNP3 também é executado pela UCP em segundo plano, com uma prioridade abaixo da aplicação de usuário, de forma cíclica a cada 50ms, sendo necessários 3 ciclos de execução do driver para realizar as requisições pendentes. Diferente do driver Cliente MODBUS, o período desta tarefa é fixo em 50ms, independente do período de polling configurado nas requisições. Desta forma, o período de polling efetivamente executado pelo driver pode apresentar uma imprecisão de até 150ms.

IEC61850 Servidor

Este protocolo está disponível para a UCP da Série Hadron Xtorm nos seus canais Ethernet. Ao selecionar esta opção no MasterTool Xtorm, a UCP passa a ser servidor da comunicação IEC61850, permitindo a conexão com clientes IEC61850 através do protocolo MMS e também possibilitando o envio e recepção de mensagens rápidas através do protocolo GOOSE.

A tabela a seguir descreve os limites máximos de configuração relacionados ao protocolo IEC 61850 Servidor.

	IEC 61850 Servidor
Instâncias por Interface Ethernet (NETs)	1
Instâncias por UCP	1
Numero máximo de Centros de Controle	5
Numero máximo de Logical Nodes	255
Número máximo de Datasets	100
Número máximo de Datasets Dinâmicos	10
Número máximo de membros em um Dataset	300
Número máximo de GOOSE Control Blocks	100
Número máximo de GOOSE Subscriptions	100
Número máximo de Report Control Blocks	100

Tabela 4-96. Limites do protocolo IEC 61850 Servidor

Para configurar este protocolo, é necessário executar os seguintes passos:

- Adicionar a instância do protocolo Servidor IEC61850 a um dos canais Ethernet disponíveis (NET 1 ao NET 6). Para realizar esse procedimento, consultar o capítulo Inserindo uma Instância de Protocolo
- Configurar a interface Ethernet. Para realizar esse procedimento, consultar o capítulo Configuração das Interfaces Ethernet
- Adicionar um Logical Device
- Adicionar os Logical Nodes desejados, incluindo objetos e atributos opcionais caso necessário
- Criar e configurar Datasets, selecionando elementos dos Logical Nodes previamente configurados
- Criar e configurar Reports (buferizados e não-buferizados), utilizando os Datasets definidos anteriormente
- Criar e configurar o envio de mensagens GOOSE (GOOSE Publisher), utilizando os Datasets definidos anteriormente
- Configurar a recepção de mensagens GOOSE (GOOSE Subscriber) através da importação de arquivos .CID
- Configurar o mapeamento de pontos de comunicação da UTR para atributos dos Logical Nodes

As descrições de cada configuração estão relacionadas a seguir, neste capítulo.

Implementação do Modelo de Dados IEC61850

Uma das principais diferenças da norma IEC 61850 quando comparada aos outros protocolos conhecidos (DNP3, MODBUS, etc...) está no modelo de dados utilizado para comunicação. Na IEC 61850, os dados são definidos sob a forma de grandes estruturas de dados simbólicos que representam um elemento físico ou lógico presente no sistema. Estas estruturas são chamadas de Logical Nodes, os quais são formados por Objetos que, por sua vez, possuem um conjunto de Atributos. Por fim, os atributos podem ser simples ou estruturados. A Figura 4-34 ilustra a estrutura hierárquica deste modelo de dados. Os Logical Nodes são representados pelos símbolos na cor laranja, enquanto os objetos possuem a cor verde e os atributos a cor azul:

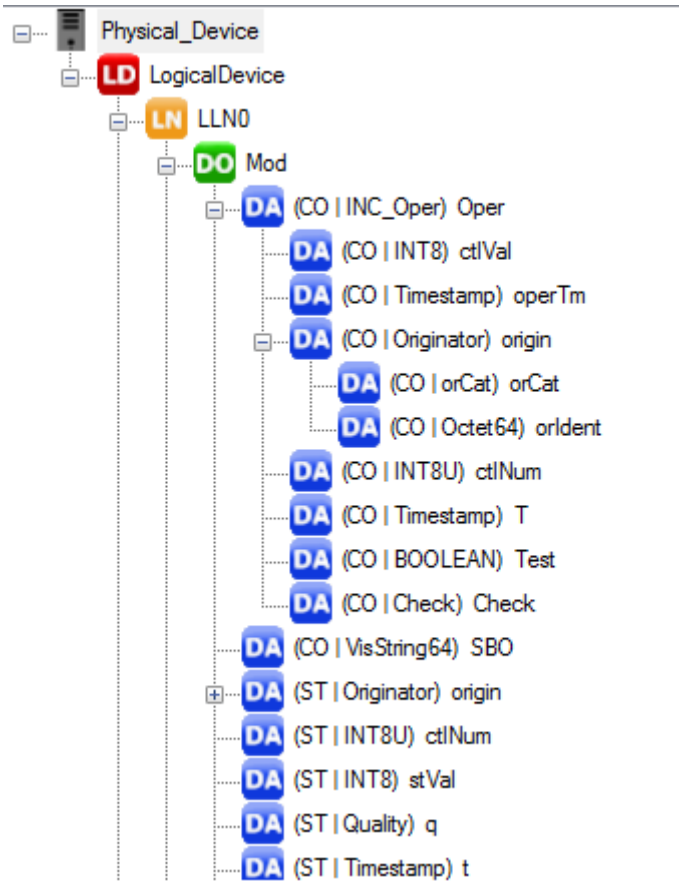


Figura 4-34. Modelo de dados estruturado

Apesar de serem representados sob a forma de uma estrutura de dados, os Logical Nodes muitas vezes possuem um algoritmo associado, o qual pode ser executado em um IED específico ou também de forma distribuída entre vários IEDs.

A implementação desta estrutura de dados IEC 61850 no ambiente de programação IEC 61131-3 do MasterTool Xtorm é realizada através de POUs do tipo Bloco Funcional. Os Logical Devices, Logical Nodes e os Objetos são representados por um Bloco Funcional, enquanto os Atributos tornam-se variáveis simbólicas de tipos elementares conforme a IEC 61131-3. Todos estes dados são declarados em uma lista de variáveis globais (GVL) criada automaticamente pela ferramenta MasterTool Xtorm e são armazenadas na pasta “IEC61850 Generated POUs” localizada na árvore do projeto. A Figura 4-35 mostra um exemplo das POUs geradas a partir de uma configuração IEC 61850:

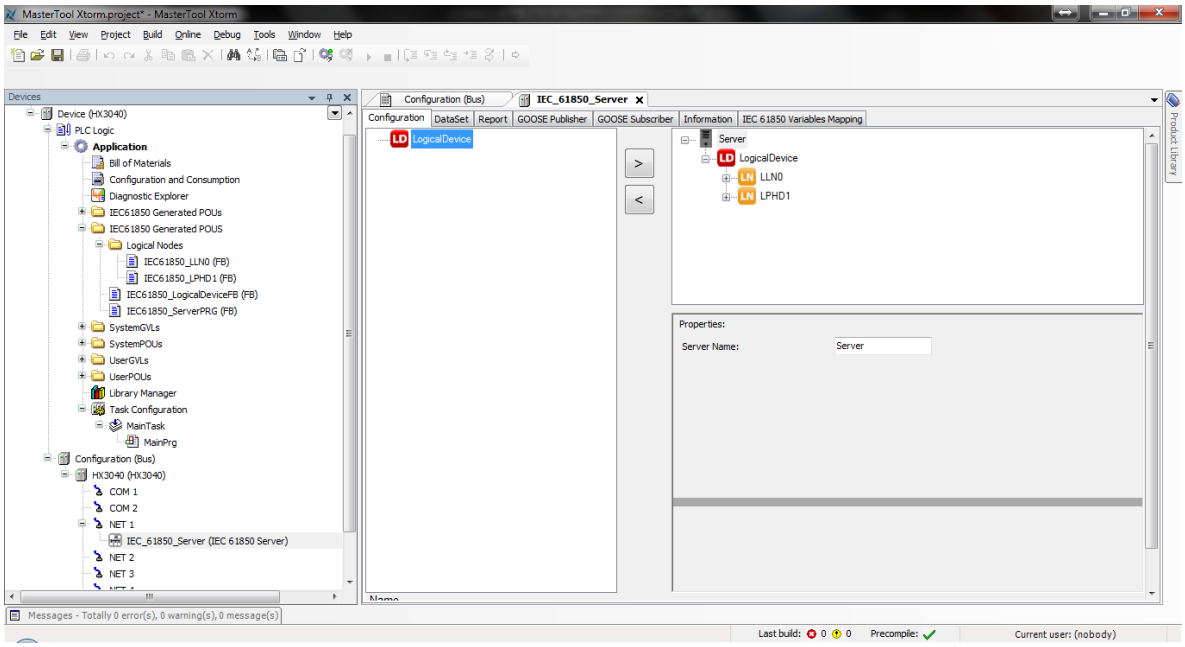


Figura 4-35. Implementação do modelo de dados

Para cada instância de um Logical Node presente na configuração IEC 61850, é criado um tipo correspondente de Bloco Funcional (armazenado na pasta “Logical Nodes”). Isto é necessário pois cada Logical Node pode ser customizado através da inclusão de Objetos e Atributos opcionais.

Conforme citado anteriormente, cada Atributo torna-se uma variável simbólica no ambiente de programação IEC 61131-3. A Tabela 4-97 mostra a correspondência entre os tipos de dados básicos da IEC 61850 com os tipos de dados elementares da IEC 61131-3:

Tipo básico IEC 61850	Tipo elementar IEC 61131-3
BOOLEAN	BOOL
INT8	SINT
INT16	INT
INT32	DINT
INT8U	USINT
INT16U	UINT
INT32U	UDINT
FLOAT32	REAL
FLOAT64	LREAL
ENUMERATED	INT
CODED ENUM	INT
OCTET STRINGnn VISIBLE STRINGnn UNICODE STRINGnn	STRING(nn)

Tabela 4-97. Correspondência dos tipos de dados IEC 61850 vs IEC 61131-3

Os tipos de dados enumerados possuem um conjunto de elementos cujos valores recebem um nome. A Tabela 4-98 a seguir descreve os valores (em base decimal) que representam na memória da UTR cada um dos nomes dos elementos:

CDC (Common Data Class)	Atributo	Identificador	Valor (decimal)
-	origin.OrCat	not-supported	0
		bay-control	1
		station-control	2
		remote-control	3
		automatic-bay	4
		automaticstation	5
		automatic-remote	6
		maintenance	7
		process	8
ACD	dirGeneral dirPhsA dirPhsB dirPhsC dirNeut	unknown	0
		forward	1
		backward	2
		both	3
BSC	ctlVal	stop	0
		lower	16384
		higher	-32768
		reseved	-16384
CMV	angRef	V	0
		A	1
		Other	2
CURVE	setCharact	ANSI Extremly Inverse	1
		ANSI Very Inverse	2
		ANSI Normal Inverse	3
		ANSI Moderate Inverse	4
		ANSI Definite Time	5
		Long-Time Extremely Inverse	6
		Long-Time Very Inverse	7
		Long-Time Inverse	8
		IEC Normal Inverse	9
		IEC Very Inverse	10
		IEC Inverse	11
		IEC Extremely Inverse	12
		IEC Short-Time Inverse	13
		IEC Long-Time Inverse	14
		IEC Definite Time	15
		Polynom 1 to 16	17 to 32
		Multiline 1 to 16	33 to 48
DPC DPS	stVal	intermediate	0
		off	16384
		on	-32768
		BAD	-16384
SEQ MV CMV	Sev range	normal	0
		high	1
		low	2
		high-high	3
		low-low	4
SEQ	phsRef	A	0

		B	1
		C	2
SEQ	T	pos-neg-zero	0
		dir-quad-zero	1
WYE	angRef	Va	0
		Vb	1
		Vc	2
		Aa	3
		Ab	4
		Ac	5
		Vab	6
		Vbc	7
		Vca	8
		Vother	9
		Aother	10

Tabela 4-98. Elementos representados na memória da UTR

Em função da flexibilidade de tamanho dos Atributos do tipo STRING, a implementação destes elementos no ambiente IEC 61131-3 é realizada de uma forma diferente de todos os outros. Ao invés de armazenar diretamente o conjunto de caracteres da string, a variável simbólica que representa o atributo armazena apenas uma referência para uma tabela onde se encontra de fato a string propriamente. Desta forma, todos os atributos do tipo string possuem um sufixo "_REF". Esta referência é na verdade o índice para acesso a uma tabela de strings implementada sob a forma de uma GVL. Existem no total 7 tabelas de string separadas por tipo:

Nome da GVL (Tabela)	Descrição
gaVisSTRING32	VISIBLE STRING32
gaVisSTRING64	VISIBLE STRING64
gaVisSTRING129	VISIBLE STRING129
gaVisSTRING255	VISIBLE STRING255
gaUCSTRING32	UNICODE STRING32
gaOCTED64	OCTET STRING64
gaOCTED255	OCTET STRING255

Tabela 4-99. Tabelas para acesso aos atributos do tipo STRING

Por exemplo, vejamos a configuração na Figura 4-36:

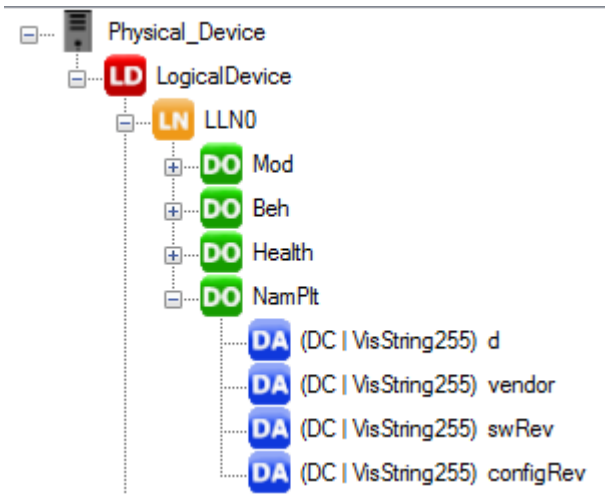


Figura 4-36. Exemplo de configuração com atributos do tipo STRING

Neste exemplo, o Atributo “d” do Objeto NamPlt é do tipo VisString255 e a sua variável simbólica correspondente é:

```
gfbIEC61850_LogicalDevice.LLN0.fbNamPlt.d_REF
```

Desta forma, para atribuir o valor “TEST” a este atributo em linguagem ST:

```
gaVisSTRING255[gfbIEC61850_LogicalDevice.GGIO4.fbNamPlt.d_REF]:= 'TEST';
```

Configuração da Base de Dados

A configuração da Base de Dados do protocolo IEC61850 Servidor é realizada através da aba Configuration a partir da inclusão de um Logical Device e de seus respectivos Logical Nodes:

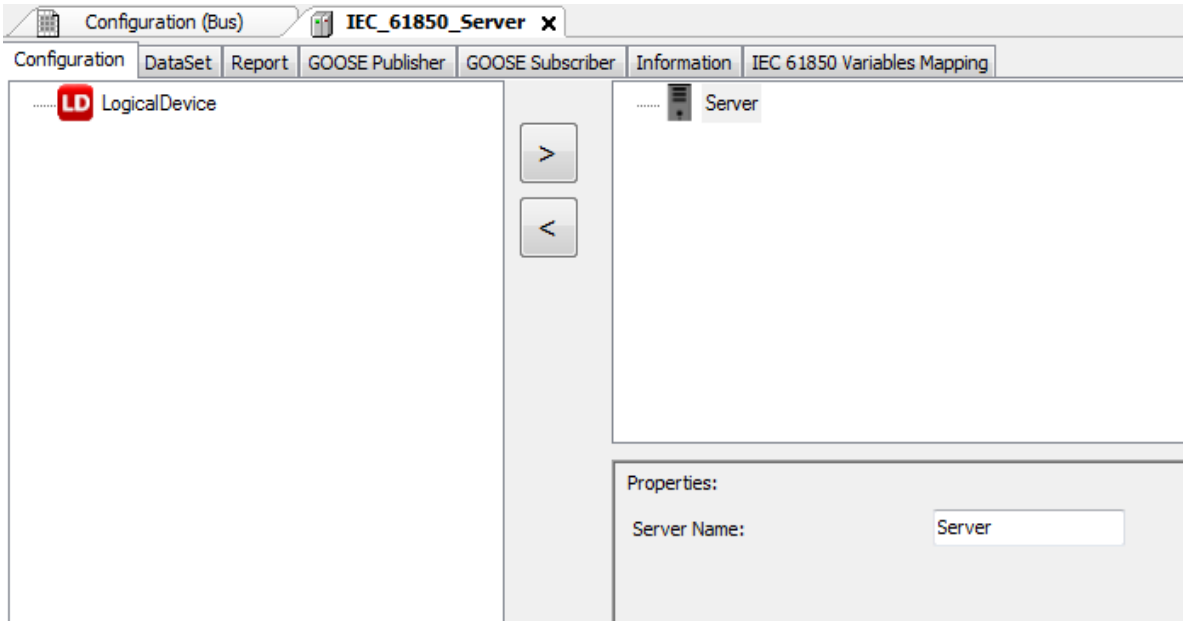


Figura 4-37. Configurando o nome do Physical Device

Esta tela é dividida em duas partes. O lado direito representa a configuração atual do dispositivo. No lado esquerdo encontra-se a lista de objetos disponíveis que podem ser adicionados/removidos conforme objeto que foi selecionado na configuração.

O primeiro passo nesta tela é definir o nome do Physical Device através do campo Server Name. Recomenda-se a utilização de um nome específico de acordo com o projeto do sistema onde esta UTR será utilizada, pois este será o nome que irá identificar este dispositivo na configuração da comunicação com dispositivos MMS Cliente e também na configuração da troca de mensagens GOOSE com outros IEDs.

O próximo passo é adicionar um Logical Device. Para isto, deve-se selecionar o objeto LogicalDevice na parte esquerda da tela e depois clicar no botão “>”:

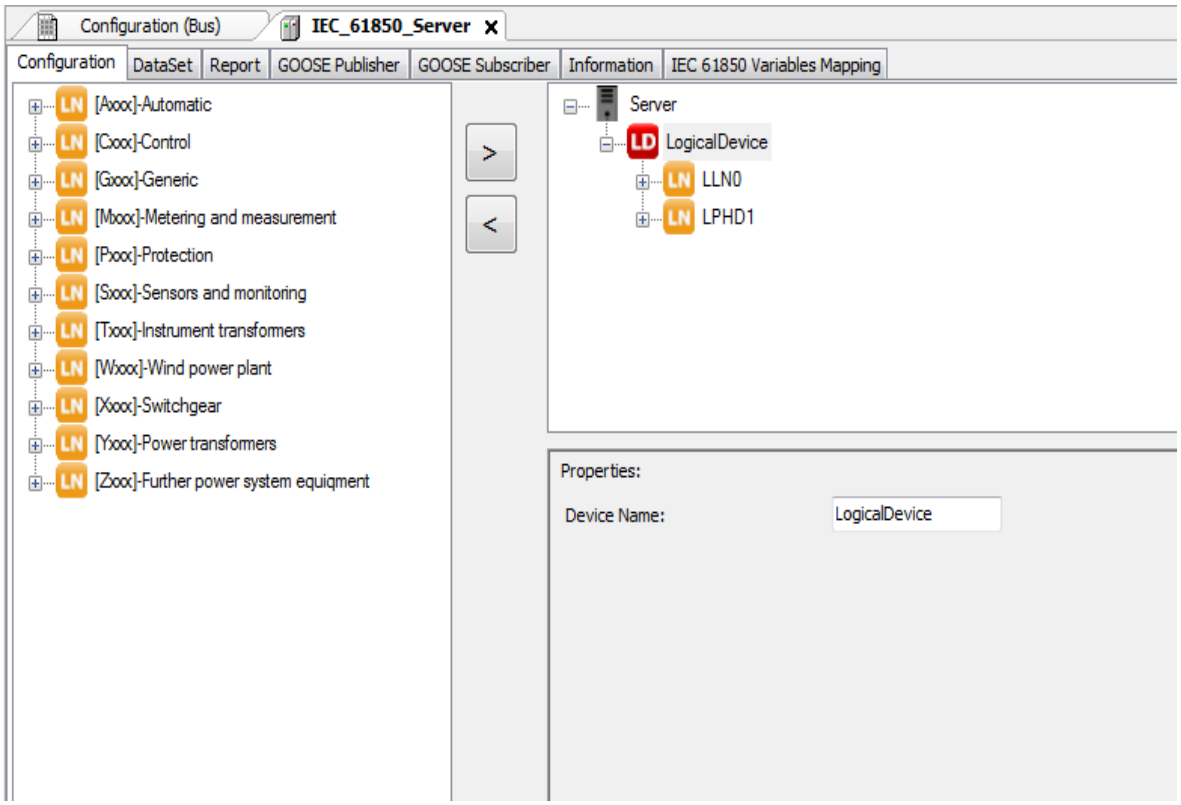


Figura 4-38. Adicionando e configurando o nome do Logical Device

O Logical Device é um nível de abstração do modelo de dados definido na IEC61850, e tem como objetivo a organização dos dados. O driver IEC61850 Servidor suporta apenas um Logical Device, e o seu nome pode ser alterado através do campo Device Name. O Logical Device vem por padrão com dois Logical Nodes que armazenam configurações e informações do dispositivo (LLN0 e LPHD1), os quais não podem ser removidos.

A partir deste ponto, o usuário deve incluir os Logical Nodes conforme a necessidade do seu projeto. Para instanciar novos Logical Nodes, basta selecioná-lo na lista que aparece do lado esquerdo da tela:

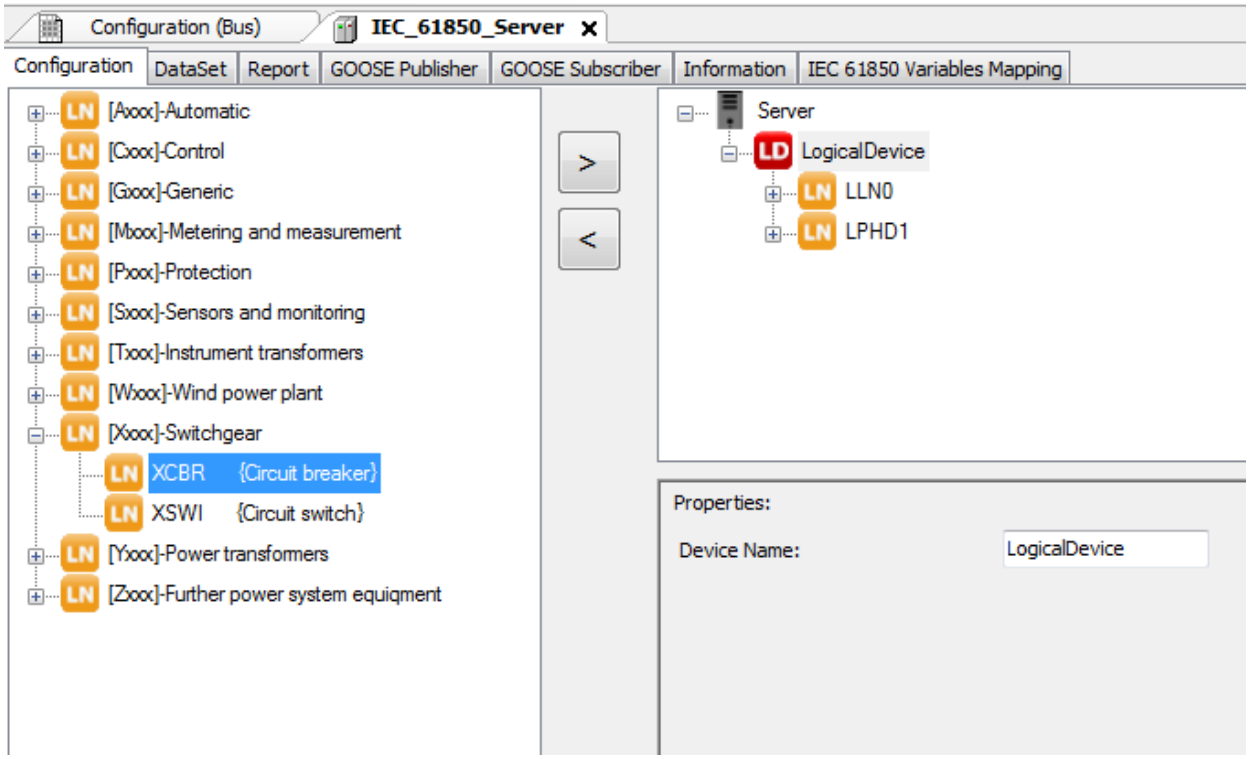


Figura 4-39. Adicionando um Logical Node

Ao clicar sobre um Logical Node configurado, é possível modificar o seu prefixo e o seu índice através dos campos Node Prefix e Logical Node Index. Além disto, é possível também adicionar e remover Data Objects opcionais. Da mesma forma, ao clicar sobre um Data Object, é possível adicionar e remover Data Attributes opcionais.

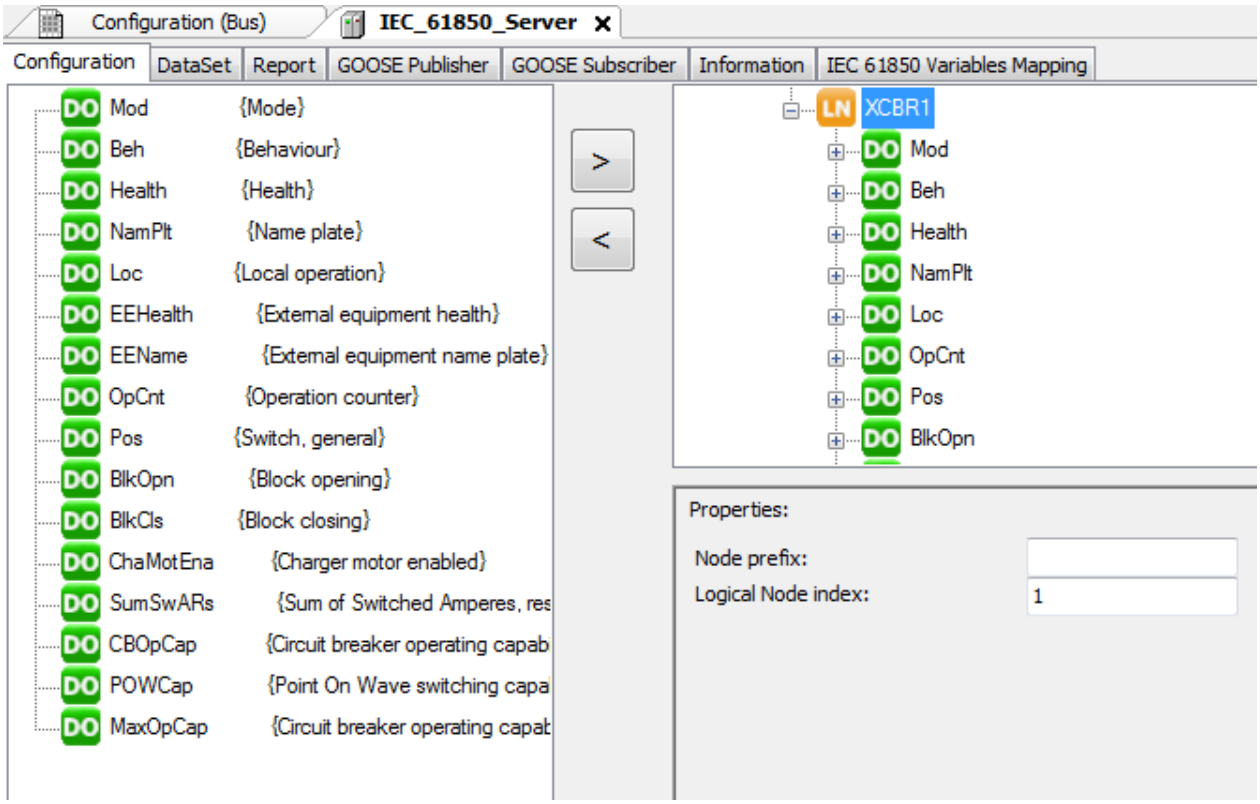


Figura 4-40. Configurando um Logical Node

A Tabela 4-100 mostra todos os campos de configuração presentes nesta tela, informando seus valores padrão e as possibilidades de configuração:

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Server Name	Define o nome do Physical Device	Server	Conjunto de até 80 caracteres (letras, números e underline)
Device Name	Define o nome do Logical Device	LogicalDevice	Conjunto de até 32 caracteres (letras, números e underline)
Node Prefix	Define o prefixo do Logical Node	-	Conjunto de caracteres (apenas letras) cuja quantidade somada ao número de caracteres do índice não pode ultrapassar 07 caracteres.
Logical Node Index	Define o índice do Logical Node	-	Conjunto de caracteres (apenas números) cuja quantidade somada ao número de caracteres do prefixo não pode ultrapassar 07 caracteres.

Tabela 4-100. Parâmetros da configuração da base de dados

Configuração dos Datasets

Os Datasets são listas de variáveis que são utilizadas na comunicação através de mensagens GOOSE e MMS (Reports, etc...). Em outras palavras, consistem em um subconjunto de dados que contém referências para Atributos e/ou Objetos da base de dados configurada no driver.

A configuração dos Datasets é realizada através da aba Dataset:

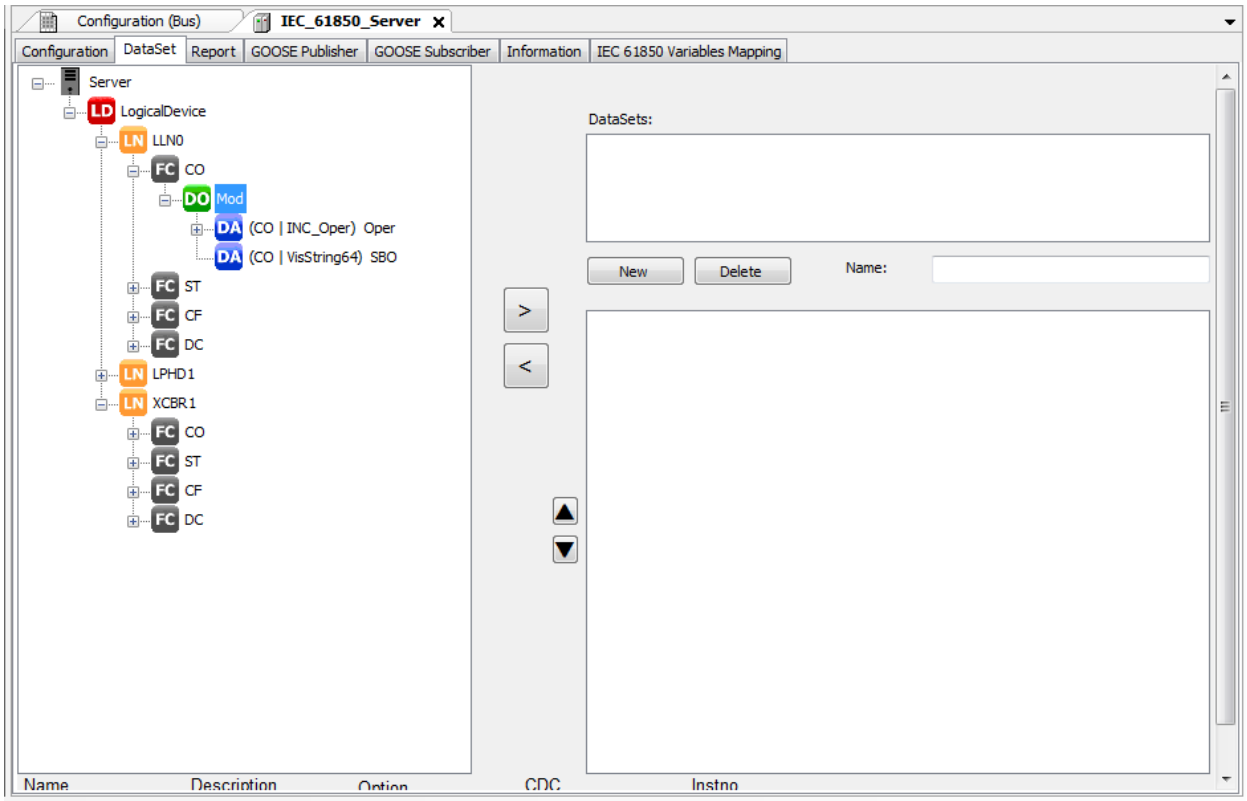


Figura 4-41. Configurando os Datasets

Esta tela é dividida em duas partes. No lado direito é apresentada a lista de Datasets atualmente configurados no dispositivo, onde existem também os comandos para criar, remover e renomear Datasets. No lado esquerdo, encontra-se a lista de elementos disponíveis que podem ser adicionados/removidos em cada Dataset a partir da base de dados que foi configurada.

Adicionalmente, a UTR Hadron Xtorm possui suporte ao recurso de Datasets Dinâmicos, o qual permite a criação de Datasets em tempo de execução pelo Cliente 61850 (SCADA, etc...). Após serem criados, é possível associá-los a um Report Control Block já existente. A configuração dos Datasets Dinâmicos é volátil, isto é, será perdida em caso de desligamento ou reconfiguração da UTR.

Configuração dos Reports

O Report é um mecanismo de envio de um Dataset de forma não solicitada para um Cliente IEC 61850. As suas configurações (Dataset associado, condições de disparo, etc...) são todas armazenadas em uma entidade chamada Report Control Block (RCB).

A configuração dos Reports é realizada através da aba Report:

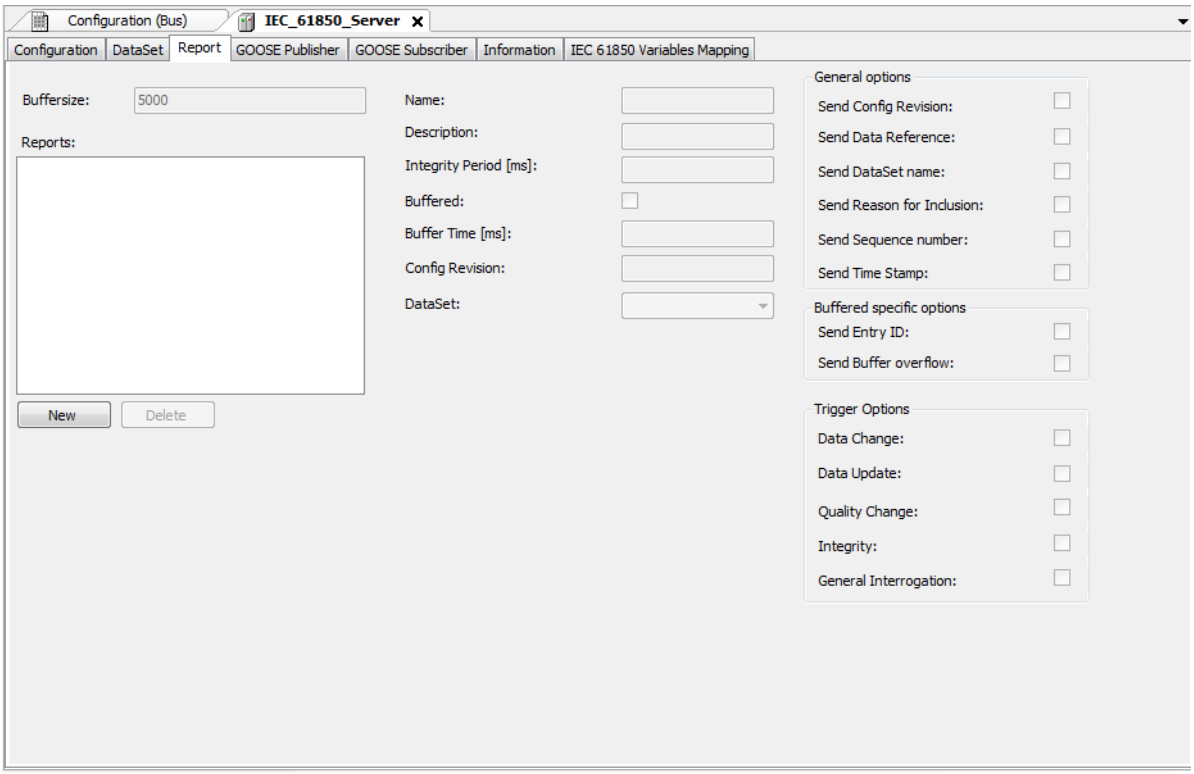


Figura 4-42. Configurando os Reports

No lado esquerdo da tela é apresentada a lista de RCBs, onde existem também os comandos para criar e remover Reports. No lado direito encontram-se as configurações do RCB.

O princípio de funcionamento do Report está baseado no envio das informações para o Cliente MMS sempre que houver qualquer alteração nos elementos contidos no Dataset daquele Control Block. Adicionalmente, o Report pode ser configurado para ser enviado periodicamente a partir do parâmetro Integrity Period. O Report pode também ser configurado no modo buferizado, onde as mensagens são armazenadas em um buffer interno da UTR para envio posterior caso o Cliente MMS não esteja conectado. Este buffer possui tamanho fixo de 20Kbytes para cada Report Control Block, sendo que a política de estouro é “manter os mais antigos” (isto é, caso o buffer esteja cheio e seja gerado um novo report, os reports armazenados no buffer serão mantidos, e o novo report será descartado).

A Tabela 4-101 mostra todos os campos de configuração presentes nesta tela, informando seus valores padrão e as possibilidades de configuração:

Configurações	Descrição	Padrão de Fábrica	Possibilidades

Buffersize	Define o tamanho (em bytes) do buffer para armazenamento dos eventos (para Buffered Reports)	5000	0 a 4294967295
Name	Nome que identificará este Report Control Block para fins de configuração e de comunicação com outros dispositivos.	RCB_n	Conjunto de até 32 caracteres (letras, números e underline)
Description	Descrição do Report Control Block, para auxiliar na documentação do projeto	-	Conjunto de até 80 caracteres (letras, números e underline)
Integrity Period (ms)	Período no qual o Report será disparado automaticamente caso nenhuma das outras condições de disparo tenha ocorrido	0	0 – Desabilitado (nunca envia) 1 a 65535
Buffered	Seleciona o modo Buferizado, no qual os Reports são armazenados internamente na UTR para envio posterior caso o Cliente MMS não esteja conectado	Desabilitado	Habilitado/Desabilitado
Buffer Time (ms)	Tempo que será aguardado após a ocorrência de um evento para que o Report seja transmitido	100	0 a 65535
Config. Revision	Número que será utilizado para identificar a revisão da configuração do Report Control Block	1	0 a 65535
Dataset	Dataset que será monitorado e transmitido por este Report Control Block	-	-
Send Config Revision	Caso selecionado, envia o parâmetro Revisão da Configuração juntamente com o Report	Habilitado	Habilitado/Desabilitado
Send Data Reference	Caso selecionado, envia a referência de dados juntamente com o Report	Habilitado	Habilitado/Desabilitado
Send Dataset Name	Caso selecionado, envia o nome do Dataset juntamente com o Report	Habilitado	Habilitado/Desabilitado
Send Reason for Inclusion	Caso selecionado, envia o motivo da inclusão juntamente com o Report	Habilitado	Habilitado/Desabilitado
Send Sequence Number	Caso selecionado, envia o número de sequência juntamente com o Report	Habilitado	Habilitado/Desabilitado
Send Time Stamp	Caso selecionado, envia a estampa de tempo juntamente com o Report	Habilitado	Habilitado/Desabilitado
Send Entry ID	Caso selecionado, envia o ID da Entrada juntamente com o Report	Desabilitado	Habilitado/Desabilitado
Send Buffer Overflow	Caso selecionado, envia a informação de Buffer Overflow juntamente com o Report	Desabilitado	Habilitado/Desabilitado
Data Change	Caso selecionado, habilita a condição de disparo do Report em função da alteração de valor	Habilitado	Habilitado/Desabilitado
Data Update	Caso selecionado, habilita a condição de disparo do Report em função da atualização do valor da variável (independente se o valor foi alterado).	Habilitado	Habilitado/Desabilitado
Quality Change	Caso selecionado, habilita a condição de disparo do Report em função da alteração de qualidade	Habilitado	Habilitado/Desabilitado
Integrity	Caso selecionado, habilita a condição de disparo do Report caso solicitado o Polling de Integridade pelo Cliente MMS	Habilitado	Habilitado/Desabilitado
General Interrogation	Caso selecionado, habilita a condição de disparo do Report em função do recebimento de um comando de Interrogação Geral enviado pelo Cliente MMS	Habilitado	Habilitado/Desabilitado

Tabela 4-101. Parâmetros de configuração dos Reports

Notas:

Buffersize: Este parâmetro não é utilizado pelo driver IEC 61850 Servidor. O tamanho de buffer é fixo em 20Kbytes para cada Buffered Report Control Block.

Data Update: Esta opção não é suportada, desta forma não é utilizada pelo driver.

General Interrogation: Esta opção não é suportada, desta forma não é utilizada pelo driver.

Configuração GOOSE

O GOOSE (Generic Object Oriented Substation Events) é uma forma de comunicação definida pela IEC 61850 criada para permitir a rápida troca de informações entre IEDs. A comunicação GOOSE utiliza o modelo Publisher/Subscriber (Produtor/Consumidor), isto é, as mensagens transmitidas por um IED são recebidas por todos os outros dispositivos presentes na rede, cabendo a eles então avaliar se desejam armazenar e utilizar os dados contidos na mensagem. As mensagens GOOSE são encapsuladas diretamente sobre o frame Ethernet utilizando endereços MAC Multicast ou Broadcast, não havendo relação com as camadas TCP/IP.

A configuração das mensagens GOOSE que serão transmitidas pela UTR Xtorm é realizada na aba GOOSE Publisher:

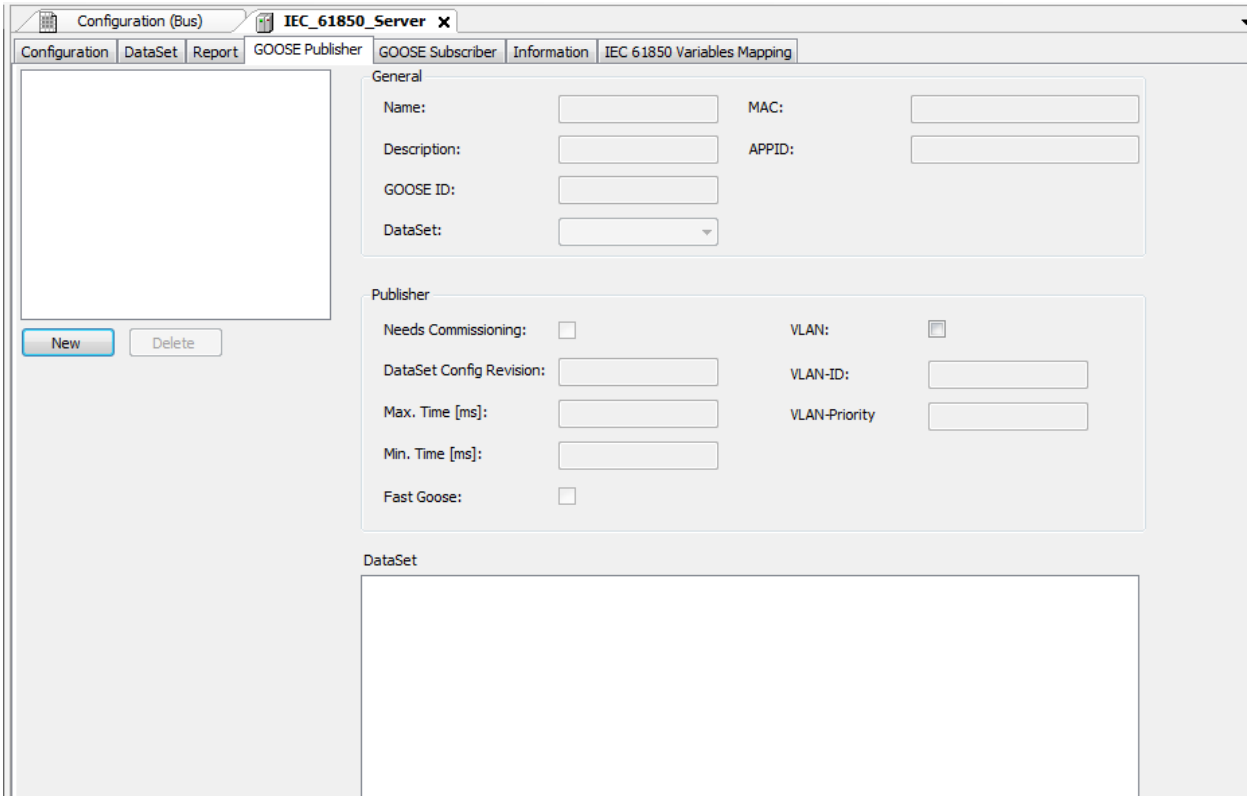


Figura 4-43. Configurando a comunicação GOOSE Publisher

As configurações de uma mensagem GOOSE a ser transmitida pela UTR Xtorm (Dataset associado, tempos de envio, etc...) são todas armazenadas em uma entidade chamada GOOSE Control Block (GCB). No lado esquerdo da tela é apresentada a lista de GCBs, onde existem também os comandos para criar e remover GCBs. No lado direito encontram-se as configurações do GCB. Na parte inferior da tela, é possível visualizar os elementos que constituem o Dataset que foi associado a esta mensagem.

O princípio de funcionamento do GOOSE Publisher está baseado no envio da mensagem quando houver qualquer alteração nos elementos contidos no Dataset daquele Control Block. Após o envio da mensagem, são realizadas retransmissões utilizando a base de tempo definida pelo parâmetro Min. Time. O intervalo de tempo para a primeira retransmissão é de 1X Min. Time,e depois vai sendo

dobrado (2X, 4X, 8X, etc...) até alcançar o intervalo definido pelo parâmetro Max. Time. Caso o Dataset não apresente alterações dentro deste tempo, a mensagem é retransmitida continuamente com o intervalo de Max. Time (conhecido também como heartbeat).

As atividades de comparação do dataset e envio da mensagem executadas por cada Control Block estão associadas a uma tarefa da UTR. Caso a opção Fast Goose esteja habilitada, o Control Block executa estas atividades na ProtTask, a qual é uma tarefa cíclica de alta prioridade com intervalo bastante curto (o default é 4ms), permitindo então configurar mensagens GOOSE extremamente rápidas para atender requisitos de desempenho exigidos por partes críticas do sistema. Caso esta opção não esteja habilitada, o Control Block executa estas atividades na tarefa principal da UTR, a MainTask.

ATENÇÃO:
Caso um Dataset esteja associado a um GOOSE Control Block cuja opção Fast Goose esteja habilitada, não é possível utilizar este mesmo Dataset em um Report Control Block. Caso este tipo de configuração seja realizada, o Report não funcionará corretamente.

A Tabela 4-102 mostra todos os campos de configuração presentes nesta tela, informando seus valores padrão e as possibilidades de configuração:

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Name	Nome que identificará este GOOSE Control Block para fins de configuração e de comunicação com outros dispositivos.	GCB_n	Conjunto de até 32 caracteres (letras, números e underline)
Description	Descrição do GOOSE Control Block, para auxiliar na documentação do projeto	-	Conjunto de até 80 caracteres (letras, números e underline)
GOOSE ID	Também é conhecido como AppID, é uma string de identificação definida pelo usuário.	GoCBRef	Conjunto de até 65 caracteres (letras, números e underline)
Dataset	Dataset que será monitorado e transmitido por este GOOSE Control Block	-	-
MAC	Endereço MAC de destino (multicast)	01-0C-CD-01-00-00	01-0C-CD-01-00-00 a 01-0C-CD-01-01-FF
APP ID	Numero de identificação da aplicação. Este número pode ser utilizado para distinguir mensagens GOOSE enviadas por diferentes partes do sistema de automação.	0x0000	0x0000 a 0x3FFF
Needs Comissioning	Deve ser habilitada pelo usuário para indicar que a configuração deste GOOSE Control Block ainda não foi concluída.	Desabilitado	Habilitado/Desabilitado
Dataset Config Revision	Número que deve ser utilizado para indicar a revisão do Dataset associado a este GOOSE Control Block	1	0 a 4294967295
Max. Time (ms)	Intervalo de tempo máximo para a retransmissão de uma mensagem GOOSE	4000	0 a 4294967295
Min. Time (ms)	Intervalo de tempo mínimo para a retransmissão de uma mensagem GOOSE	500	0 a 4294967295
Fast GOOSE	Define em qual tarefa da UTR será executado este Control Block. Caso selecionado, a comparação dos valores e envio da mensagem GOOSE serão executados na ProtTask. Caso contrário, esta atividade será realizada na MainTask.	Desabilitado	Habilitado/Desabilitado
VLAN	Caso selecionado, habilita a utilização de VLAN na mensagem GOOSE	Desabilitado	Habilitado/Desabilitado

VLAN ID	Número de identificação da VLAN	0x0000	0x0000 a 0x0FFF
VLAN Priority	Número prioridade da VLAN	4	0 a 7

Tabela 4-102. Parâmetros de configuração do GOOSE Publisher

A configuração das mensagens GOOSE que serão recebidas pela UTR Xtorm é realizada na aba GOOSE Subscriber através da importação de arquivos SCL (Substation Configuration Language). Estes arquivos são gerados pela ferramenta de configuração do IED responsável por enviar a mensagem, e armazenam internamente as informações dos GOOSE Control Blocks relacionados. A importação é realizada através do botão localizado no canto superior esquerdo da tela:

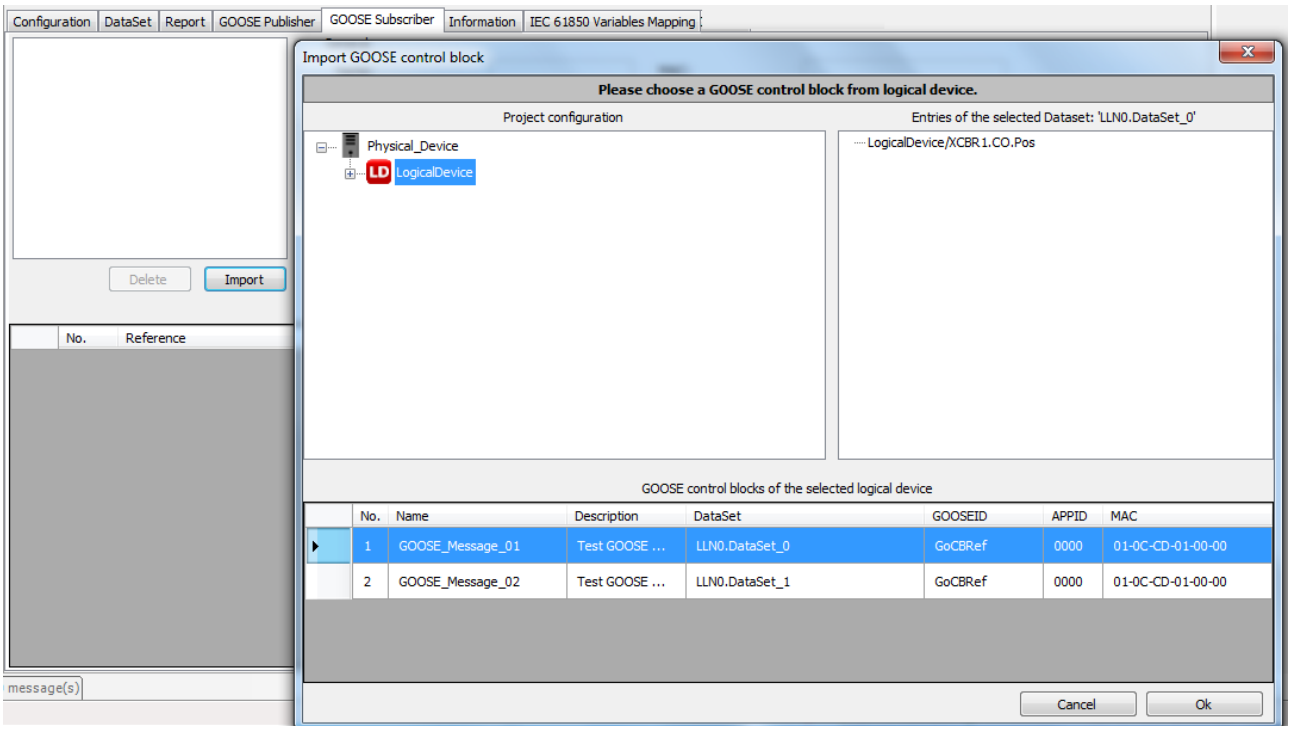


Figura 4-44. Importando um arquivo SCL

Após clicar no botão de importação e selecionar o arquivo SCL desejado, é aberta uma tela onde é apresentada uma lista de Physical Devices e Logical Devices contidos neste arquivo. Ao clicar sobre um Logical Device, é apresentada na parte inferior da tela a lista de mensagens GOOSE (Control Blocks) disponíveis para serem recebidos. Após selecionar a mensagem que se deseja receber, basta clicar em OK.

Retornando à aba GOOSE Subscriber, no canto superior esquerdo é apresentada a lista de mensagens configuradas para serem recebidas.

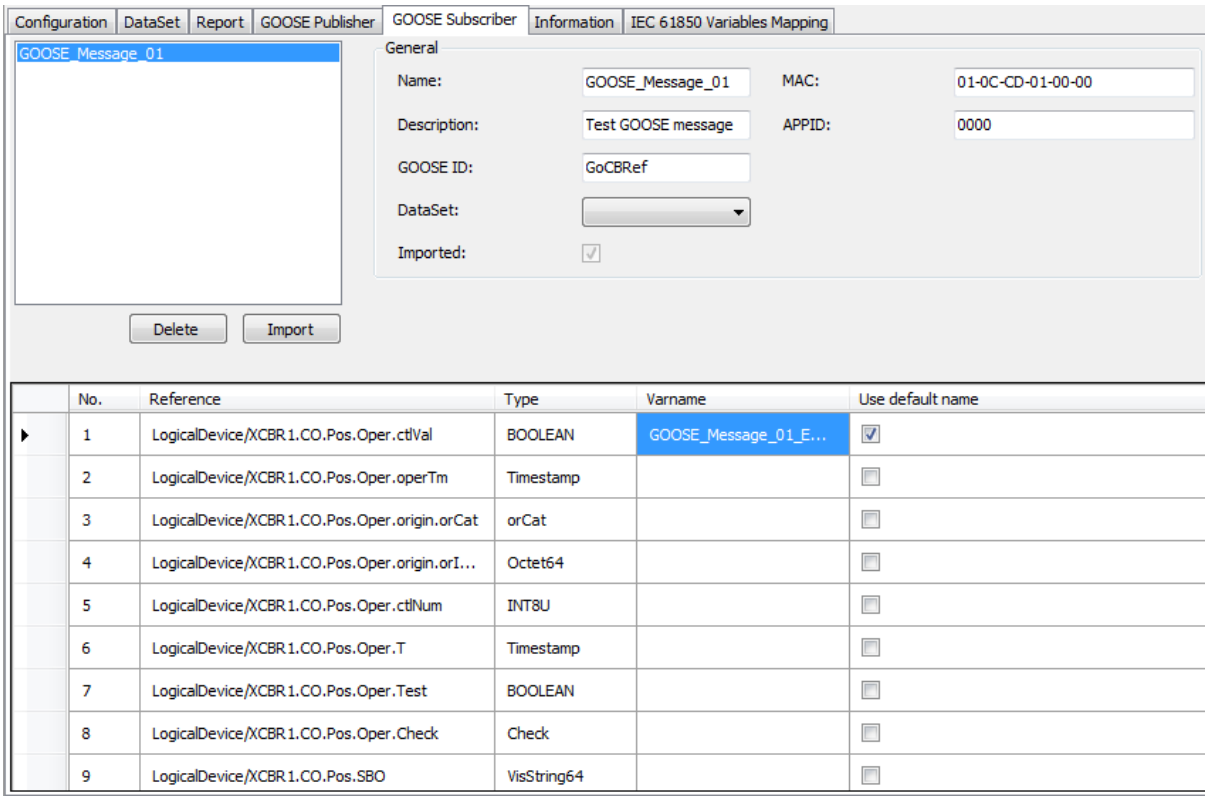


Figura 4-45. Configurando a comunicação GOOSE Subscriber

Ao selecionar uma mensagem, no lado direito são apresentadas as informações relacionadas à mensagem que será recebida. Estes campos são preenchidos automaticamente através da importação dos dados contidos no GOOSE Control Block associado à mensagem, desta forma não devem ser modificados pelo usuário.

Na parte inferior desta tela são apresentados os dados contidos na mensagem selecionada, onde é possível configurar quais elementos do dataset recebido serão efetivamente armazenados na memória da UTR. Este mapeamento é realizado através do campo Varname, onde deve ser informado o nome da variável para a qual será copiado o dado no momento da recepção da mensagem. Este campo pode ser preenchido manualmente pelo usuário com o nome de uma variável já existente na sua aplicação, ou também pode ser preenchido automaticamente através da opção Use default name. Ao marcar esta opção, a ferramenta MasterTool Xtorm cria automaticamente uma variável na GVL IEC61850_Generated_GVL para armazenar o elemento correspondente recebido pela mensagem GOOSE. O nome desta variável é formado pelo nome do GOOSE Control Block adicionando o sufixo "_EntryX", onde X é um número que indica a posição do elemento na mensagem.

Configuração do Mapeamento de Variáveis IEC 61850

A aba IEC 61850 Variables Mapping permite realizar a associação entre variáveis internas da UTR e Atributos dos Logical Nodes. Este recurso é essencial para permitir a implementação de Logical Nodes a partir de dados oriundos de outros protocolos de comunicação, uma vez que os tipos de dados entre os protocolos podem não ser totalmente compatíveis.

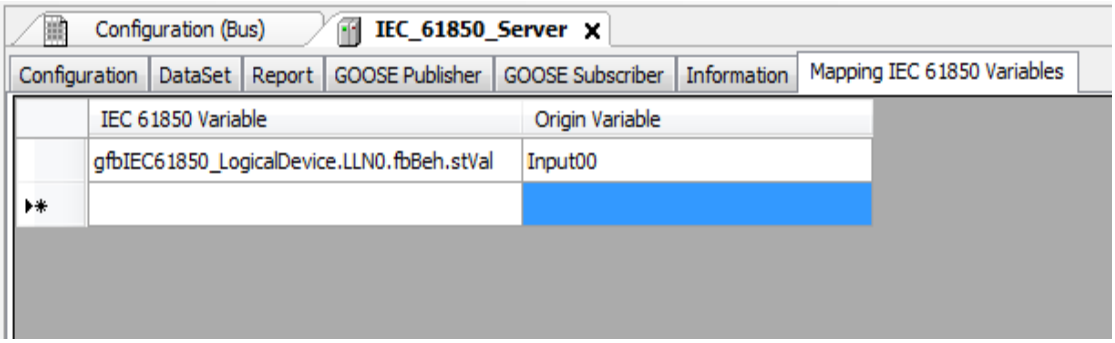


Figura 4-46. Mapeamento de Variáveis IEC 61850

A configuração do mapeamento de Variáveis IEC 61850 consiste basicamente em uma tabela com duas colunas, uma contendo a variável que representa o Atributo do Logical Node (IEC 61850 Variable) e a outra contendo a variável da UTR com a qual o Atributo será associado (Origin Variable). No exemplo da figura acima, o Atributo stVal do objeto Beh do Logical Node LLN0 foi mapeado para uma variável chamada DigitalInput00. Visto que esta associação pode implicar em uma conversão de tipo de dado, uma Variável Origem pode somente ser associada a um Atributo cujo tipo tenha a mesma sinalização (signed/unsigned) e tamanho (em bits) igual ou maior. O único tipo que não segue esta regra é o DBP, pois consiste em uma estrutura de duas variáveis BOOL mas mesmo assim pode ser mapeado para Atributos que representam o valor de pontos duplos e possuem 8 bits. A Tabela 4-103 mostra as possibilidades de mapeamento entre uma Variável Origem e uma Variável IEC 61850:

Tipo da Variável Origem	Tipo da Variável IEC 61850	
	IEC 61131-3	IEC 61850
BOOL	BOOL	BOOLEAN
SINT	SINT, DBP	INT8
INT	INT, DINT, REAL	INT16, INT32, REAL
DINT	DINT, REAL	INT32, REAL
USINT	USINT	INT8U
UINT	DINT, UDINT, REAL	INT32, INT32U, REAL
UDINT	UDINT,REAL	INT32U, REAL
REAL	REAL	FLOAT32
DBP	SINT	INT8

Tabela 4-103. Tipos de dados possíveis no mapeamento de Variáveis IEC 61850

Caso a Variável Origem esteja associada a uma fonte capaz de gerar eventos com estampa de tempo própria (IED DNP3 ou módulo de entrada digital por exemplo), o driver IEC 61850 Server receberá estes eventos e atualizará o atributo “t” (Timestamp) do Objeto referente ao atributo que foi mapeado, se comportando como um “consumidor” conforme descrito na seção Fila de Eventos da UCP.

Diagnósticos IEC 61850 Servidor

Os diagnósticos do protocolo IEC 61850 Servidor configurado são armazenados em variáveis do tipo T_DIAG_IEC61850_SERVER. As informações relacionadas à transmissão/recepção de mensagens GOOSE são armazenadas em arrays com dimensão correspondente à quantidade máxima que pode ser configurada. Cada posição do array possui uma estrutura que possui informações correspondes a um GOOSE Control Block ou a um Subscription. Os diagnósticos estão descritos na Tabela 4-104:

Variável de diagnóstico do tipo T_DIAG_IEC61850_SERVER.*	Tamanho	Descrição
Diagnósticos		

bDriverConfigured			BOOL	Indica que o driver foi configurado corretamente
tGooseGenDiagnostic		uiGooseControlBlocks	UINT	Número de GOOSE Control Blocks (GCBs) configurados
		uiGooseSubscriptions	UINT	Número de mensagens GOOSE configuradas para serem recebidas
tGooseControlBlock	**tGooseControlBlock[n]**	bActive	BOOL	Indica que esta estrutura de diagnósticos está ativa
		uiAPPID	UINT	Número de identificação do GOOSE Control Block
		udiDataChange	UDINT	Número de vezes que foi detectada alguma alteração nos elementos do Dataset
		udiRetransmission	UDINT	Número de retransmissões da mensagem GOOSE (é zerado sempre que houver um Data Change)
		udiTotal	UDINT	Total de mensagens transmitidas (inclui a transmissão inicial e todas as retransmissões)
tGooseSubscription	**tGooseSubscription[n]**	bActive	BOOL	Indica que esta estrutura de diagnósticos está ativa
		uiAPPID	UINT	Número de identificação do GOOSE Control Block correspondente à mensagem configurada para recepção
		udiReceived	UDINT	Número de recepções da mensagem GOOSE configurada para recepção
		udiDataChange	UDINT	Número de vezes que foi detectada alguma alteração nos elementos do Dataset ao receber a mensagem
		udiTimeout	UDINT	Indica que não houve recepção da mensagem GOOSE dentro do tempo definido por Time Allowed to Live (TAL)
		tTimestamp	T_TIMESTAMP	Estampa de tempo do momento em que detectada alguma alteração nos elementos do Dataset ao receber a mensagem
tReportControlBlock	**tReportControlBlock[n]**	sRCB_NAME	STRING(65)	Nome do Report Control Block
		sIP_ADDR	STRING(15)	Endereço IP do Cliente 61850 que habilitou e/ou reservou este Report
		bEnabled	BOOL	Indica que este Report foi habilitado por um Cliente 61850
		bResv	BOOL	Indica que este Report foi reservado por um Cliente 61850 (aplicável somente para Buffered Reports)
		bBuffered	BOOL	Indica que este Report é do tipo Buffered
		byOccupation	BYTE	Informa a ocupação do buffer interno (de 0 a 100, em %, aplicável somente para Buffered Reports)
tConnectionStatus	**tConnectionStatus[n]**	bActive	BOOL	Indica que há conexão estabelecida com um Cliente 61850
		sIP_ADDR	STRING(15)	Endereço IP do Cliente 61850 que está conectado
		udiFramesTx	UDINT	Informa a quantidade de frames transmitidos na comunicação com o Cliente 61850 que está conectado
		udiFramesRx	UDINT	Informa a quantidade de frames recebidos na comunicação com o Cliente 61850 que está conectado

Tabela 4-104. Diagnósticos IEC 61850 Servidor

Interceptação de Comandos

A interceptação de comandos recebidos pelo driver IEC 61850 Servidor é realizada através do bloco de função CommandReceive conforme descrito na seção Interceptação de Comandos Oriundos do Centro de Controle. Este recurso permite a interceptação de comandos de seleção e operação para os seguintes CDCs (Common Data Classes): SPC, DPC, INC, BSC, ISC.

Abaixo um exemplo de aplicação utilizando a linguagem ST mostrando a interceptação de um comando para o objeto BlkOpn de um Logical Node XCBR, respondendo "SUCCESS" ao Cliente IEC 61850.

```
PROGRAM UserPrg
VAR
   bSetup: BOOL:= TRUE;
   CRReceive: CommandReceiver;
   CComamnd: COMMAND_T;
   CRResult: COMMAND_RESULT:= COMMAND_RESULT.SUCCESS;
END_VAR

IF bSetup THEN // Executa o Setup da função no primeiro ciclo
   CRReceive.dwVariableAddr:=ADR(
gfbIEC61850_LogicalDevice.XCBR1.fbBlkOpn.Oper.ctlVal);
   CRReceive.bExec:= TRUE;
   CRReceive.eCommandResult:= COMMAND_RESULT.NONE;
   CRReceive.dwTimeout:= 8000;
   bSetup:= FALSE;
RETURN;
END_IF

IF CRReceive.bDone THEN
   CRReceive.bDone:= FALSE;
END_IF

CRReceive();

IF CRReceive.bCommandAvailable THEN // Caso um comando seja capturado:
CComamnd:= CRReceive.sCommand; // Salva os dados do comando
   CRReceive.eCommandResult:= CRResult; // Envia resposta "SUCCESS"
   CRReceive.bDone:= TRUE;
END_IF
CRReceive();
```

Desempenho do Sistema

Em casos onde a aplicação possui apenas uma tarefa de usuário MainTask, responsável pela execução de uma única unidade de programação do tipo Programa denominada MainPrg, o UCP consome um determinado tempo para que a tarefa seja processada. A esse tempo damos o nome de Tempo de Execução.

Em uma aplicação, podemos conhecer o Tempo de Execução médio da aplicação usando o MasterTool Xtorm, na Árvore de Dispositivos, no item Device, no seguinte caminho:

Device-> Application-> Task Configuration na aba Monitor, na coluna Tempo de Ciclo Médio.

Deve-se atentar ao Tempo de Execução para que ele não ultrapasse 80% do intervalo configurado na tarefa de usuário MainTask. Por exemplo, em uma aplicação onde o intervalo é de 100 ms, um Tempo de Execução adequado é de até 80 ms. Isso se deve ao fato de que a UCP necessita de um tempo para a execução de outras tarefas como o processamento da comunicação, tratamento do visor

e cartão de memória, e essas tarefas também acontecem dentro do intervalo (os 20% restantes do Tempo de Execução).

O MasterTool Xtorm tem ainda a opção de limitar o Tempo de Execução da tarefa de usuário MainTask, para que o Tempo de Execução mencionado acima seja respeitado. Isto é feito consistindo o Cão-de-Guarda da tarefa de usuário em 80% do intervalo.

Caso o usuário esteja ciente das informações acima e queira utilizar esta consistência do Cão-de-Guarda, isso será possível marcando o checkbox Gerar erro na consistência do cão-de-guarda das tarefas na aba de configuração da UCP. Porém, o recomendado é manter o tempo de cão-de-guarda configurado com o padrão de 1000 ms e o checkbox para gerar erro na consistência desmarcado.

ATENÇÃO: Caso o percentual de 80% não seja respeitado e o tempo de execução da tarefa de usuário se aproxime ou exceda o intervalo configurado da MainTask, o visor e o botão de diagnóstico podem não responder, uma vez que sua prioridade na execução do sistema é menor do que as tarefas de usuário. Caso uma aplicação com erros seja carregada na UCP, pode ser necessário reinicializá-la sem carregar esta aplicação, como descrito na seção Não Carregar a Aplicação na Inicialização.

Escaneamento de E/S

A atualização das entradas e saídas associadas aos módulos presentes no barramento ocorre primariamente na tarefa principal da UTR, a MainTask, conforme a sequência descrita a seguir:

- Leitura das Entradas
- Escrita das Saídas
- Execução da aplicação do usuário
- Aguarda completar o intervalo configurado para a MainTask

Conforme descrito acima, fica claro que o Tempo de Execução será resultante da soma de 3 parcelas: Leitura das Entradas, Escrita das Saídas e Execução da aplicação do usuário. Desta forma, o Tempo de Execução da aplicação é afetado pela quantidade de módulos de entrada e saída presentes no barramento. A duração da parcela relacionada à atualização de E/S é de aproximadamente 1 ms para cada 20 módulos inseridos no projeto. A última etapa da MainTask (aguarda completar o intervalo configurado) é onde os drivers de comunicação e todos os outros serviços de baixa prioridade executam.

Adicionalmente, é possível configurar a UTR para realizar a atualização de entradas e saídas dos módulos em outras tarefas através da opção “Habilita atualização de E/S por tarefa”, localizada nas configurações da UCP. Ao habilitar esta opção, a UCP realiza um acesso ao barramento local no contexto de cada tarefa para atualização das entradas e saídas que estiverem sendo utilizadas nela. Esta opção pode ser muito útil em casos onde utiliza-se a tarefa ProtTask e seja necessário uma rápida atuação em entradas e saídas (lógicas de proteção por exemplo). A ordem das ações executadas é levemente diferente da MainTask, conforme descrito a seguir:

- Execução da aplicação do usuário
- Escrita das Saídas
- Leitura das Entradas
- Aguarda completar o intervalo configurado para a tarefa

Cartão de Memória

A transferência de dados envolvendo o cartão de memória é realizada em segundo plano pela UCP, pois esta dá prioridade para a execução da aplicação de usuário e processamento da comunicação. Desta forma, a transferência de arquivos para o cartão poderá sofrer um acréscimo de tempo significativo, dependendo do tempo de execução da aplicação do usuário.
O tempo necessário para ler/escrever arquivos no cartão será afetado diretamente pelo tempo de execução da aplicação de usuário, uma vez que essa aplicação tem prioridade na execução.

Para mais informações sobre a utilização do cartão de memória, ver capítulo Configuração - Cartão de Memória.

Relógio RTC

A UCP da Série Hadron Xtorm possuem um relógio interno que pode ser utilizado através da biblioteca LibPlcStandard.lib. Essa biblioteca é carregada automaticamente durante a criação de um novo projeto (para realizar o procedimento de inserção de uma biblioteca, consultar capítulo Programação Inicial - Bibliotecas). A Figura 4-47 mostra como incluir os blocos e funções disponíveis na biblioteca para acessar o relógio:

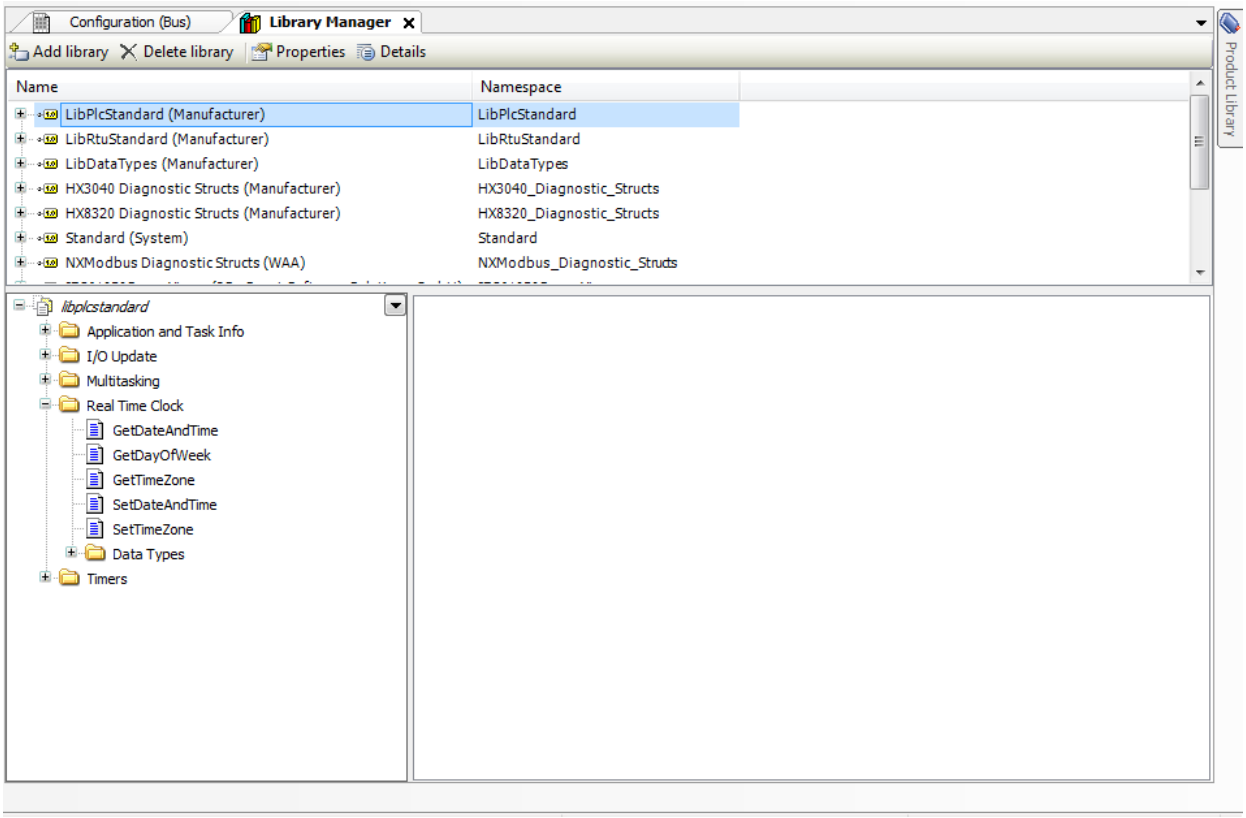


Figura 4-47. Blocos de Escrita e Leitura do Relógio

Blocos Funcionais e Funções para Leitura e Escrita do RTC

Entre outros blocos funcionais e funções, existem quatro muito importantes utilizados para a leitura do relógio (GetDateAndTime e GetTimeZone) e para configurar novos valores de data e hora (SetDateAndTime e SetTimeZone). A seguir, é descrito como utilizar os blocos e funções.

Funções de Leitura do RTC

GetDateAndTime

A função GetDateAndTime é utilizada para fazer a leitura da data e hora com o retorno em milissegundos.

Parâmetros de entrada e saída	Tipo	Descrição
DATEANDTIME	EXTENDED_DATE_AND_TIME	Recebe os valores de data e hora com milissegundos. Ver seção Estruturas de Dados do RTC - EXTENDED_DATE_AND_TIME

Tabela 4-105. Parâmetros de Entrada e Saída

Parâmetros de saída	Tipo	Descrição
GetDateAndTime	RTC_CMD_STATUS	Retorna o resultado da leitura/configuração. Ver seçãoRTC_CMD_STATUS.

Tabela 4-106. Parâmetros de Saída

Quando chamada, a função irá ler o valor atual do relógio e preencher a estrutura DATEANDTIME. O resultado da leitura é retornado pela função.

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
    xStatus     : RTC_CMD_STATUS;
    DateAndTime : EXTENDED_DATE_AND_TIME;
    xRead       : BOOL;
END_VAR
-------------------------------------------------------------------------------
IF ( xRead = TRUE) THEN
    xStatus := GetDateAndTime( DateAndTime);
    xRead := FALSE;
END_IF
```

GetDayOfWeek

A função GetDayOfWeek é utilizada para fazer a leitura do dia da semana.

Parâmetros de saída	Tipo	Descrição
GetDayOfWeek	DAYS_OF_WEEK	Retorna o dia da semana. Ver seção Estruturas de Dados do RTC - DAYS_OF_WEEK

Tabela 4-107. Parâmetros de Saída

Quando chamada, a função irá ler o dia da semana e preencher a estrutura DAYS_OF_WEEK.

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
    DayOfWeek   : DAYS_OF_WEEK;
    xRead       : BOOL;
END_VAR
-------------------------------------------------------------------------------
IF ( xRead = TRUE) THEN
    DayOfWeek := GetDayOfWeek();
    xRead := FALSE;
END_IF
```

GetTimeZone

A função GetTimeZone é utilizada para fazer a leitura das configurações de fuso horário.

Parâmetros de entrada e saída	Tipo	Descrição
TIMEZONE	TIMEZONESETTINGS	Recebe os valores de data e hora com milissegundos. Ver seção Estruturas de Dados do RTC - EXTENDED_DATE_AND_TIME

Tabela 4-108. Parâmetros de Entrada e Saída

Parâmetros de saída	Tipo	Descrição
GetTimeZone	RTC_CMD_STATUS	Retorna o resultado da leitura/configuração. Ver seção RTC_CMD_STATUS

Tabela 4-109. Parâmetros de Saída

Quando chamada, a função irá ler a configuração atual de fuso horário e preencher a estrutura TIMEZONE. O resultado da leitura é retornado pela função.

```
PROGRAM UserPrg
VAR
    RTCStatus    : RTC_CMD_STATUS;
    TimeZone     : TIMEZONESETTINGS;
    xRead        : BOOL;
END_VAR
-------------------------------------------------------------------------------
//FB GetTimeZone
IF ( xRead = TRUE) THEN
    GetTimeZone ( TimeZone);
    xRead := FALSE;
END_IF
```

Blocos Funcionais e Funções de Escrita e Configuração do RTC

SetDateAndTime

O Bloco Funcional SetDateAndTime é utilizado para realizar a configuração do relógio:

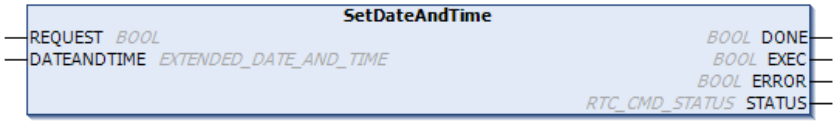


Figura 4-48. Configura Data e Hora com Milissegundos

Parâmetros de entrada	Tipo	Descrição
REQUEST	BOOL	Essa variável, quando recebe uma borda de subida, habilita a leitura do relógio.
DATEANDTIME	EXTENDED_DATE_AND_TIME	Estrutura a ser preenchida com os valores de data e hora com milissegundos. Ver seção Estruturas de Dados do RTC - EXTENDED_DATE_AND_TIME

Tabela 4-110. Parâmetros de Entrada

Parâmetros de saída	Tipo	Descrição
DONE	BOOL	Essa variável, quando verdadeira, indica que a ação foi completada com sucesso.
EXEC	BOOL	Essa variável, quando verdadeira, indica que a função está processando os valores.
ERROR	BOOL	Essa variável, quando verdadeira, indica que ocorreu algum erro durante a Escrita.
STATUS	RTC_CMD_STATUS	Retorna o erro ocorrido durante a leitura/configuração. Ver seção RTC_CMD_STATUS.

Tabela 4-111. Parâmetros de Saída

Quando ocorrer uma borda de subida na entrada REQUEST, o bloco funcional irá escrever o novo valor DATEANDTIME no relógio. Caso a escrita seja realizada com sucesso, a saída DONE será igual a TRUE. Caso contrário, a saída ERROR será igual a TRUE e o erro será apresentado na variável STATUS.

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
    SetDateAndTime    : SetDateAndTime;
    xRequest          : BOOL;
    DateAndTime       : EXTENDED_DATE_AND_TIME;
    xDone             : BOOL;
    xExec             : BOOL;
    xError            : BOOL;
    RTCStatus         : RTC_CMD_STATUS;
    xWrite            : BOOL;
END_VAR
------------------------------------------------------------------------
IF ( xWrite = TRUE) THEN
    SetDateAndTime(
    Request := xRequest,
    DateAndTime := DateAndTime,
    Done => xDone,
    Exec => xExec,
    Error => xError,
    Status => RTCStatus);
    xWrite := FALSE;
END_IF
```

ATENÇÃO:
Se o usuário tentar escrever valores de hora fora do intervalo do RTC, os valores serão convertidos para valores válidos, desde que não ultrapasse a faixa válida de 01/01/2000 até 31/12/2035. Por exemplo, se o usuário tentar escrever o valor 2000 ms, o mesmo será convertido para 2 segundos, se escrever o valor 100 segundos, o mesmo será convertido para 1 min e 40 segundos. Se escrever o valor de 30 horas, o mesmo será convertido para 1 dia e 6 horas, e assim por diante. Existe uma exceção para a geração de erros, para valores “0” dos campos de entrada de valores, a função não retornará erro. Neste caso, a data será convertida para a data anterior mais próxima. Por exemplo: se o usuário escrever a data 00/04/2014, a função irá converter para 31/03/2014.

SetTimeZone

A função SetTimeZone realiza a escrita das configurações de fuso horário:

Parâmetros de entrada	Tipo	Descrição
TIMEZONE	TIMEZONESETTINGS	Estrutura com o valor de fuso horário a ser configurado. Ver seção Estruturas de Dados do RTC - TIME ZONE SETTINGS.

Tabela 4-112. Parâmetros de Entrada

Parâmetros de saída	Tipo	Descrição
SetTimeZone	RTC_CMD_STATUS	Retorna o erro ocorrido durante a configuração. Ver seçãoRTC_CMD_STATUS.

Tabela 4-113. Parâmetros de Saída

Quando chamada, a função irá configurar o valor de TIMEZONE como a nova configuração de fuso horário do sistema. O resultado da configuração é retornado pela função.

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
   RTCStatus   : RTC_CMD_STATUS;
   TimeZone    : TIMEZONESETTINGS;
   xWrite      : BOOL;
END_VAR
-------------------------------------------------------------------------
//FB SetTimeZone
IF ( xWrite = TRUE) THEN
   RTCStatus := SetTimeZone ( TimeZone);
   xWrite := FALSE;
END_IF
```

ATENÇÃO:
Para realizar o acerto do relógio, devem-se utilizar valores de hora e datas dentro da seguinte faixa válida: -12h59min até 13h59min, caso contrário, será reportado um erro através do parâmetro de saída STATUS. Para maiores detalhes do parâmetro de saída STATUS, consultar a seção RTC_CMD_STATUS.

Estruturas de Dados do RTC

Os blocos funcionais de leitura e configuração do RTC da UCP da Série Hadron Xtorm utilizam as seguintes estruturas de dados na sua configuração:

EXTENDED_DATE_AND_TIME

Esta estrutura é utilizada para armazenar a data do RTC quando utilizados os blocos funcionais para leitura/configuração do horário:

Estrutura	Tipo	Variável	Descrição
EXTENDED_DATE_AND_TIME	BYTE	byDayOfMonth	Armazena o dia do mês da data configurada.
	BYTE	ByMonth	Armazena o mês da data configurada.
	WORD	wYear	Armazena o ano da data configurada.
	BYTE	byHours	Armazena a hora da data configurada.
	BYTE	byMinutes	Armazena os minutos da data configurada.
	BYTE	bySeconds	Armazena os segundos da data configurada.
	WORD	wMiliseconds	Armazena os milissegundos da data configurada.

Tabela 4-114. EXTENDED_DATE_AND_TIME

DAYS_OF_WEEK

Esta estrutura é utilizada para armazenar o dia da semana quando utilizada a função para leitura do dia da semana:

Enumerador	Valor	Descrição
DAYS_OF_WEEK	0	INVALID_DAY
	1	SUNDAY
	2	MONDAY
	3	TUESDAY
	4	WEDNESDAY
	5	THURSDAY
	6	FRIDAY
	7	SATURDAY

Tabela 4-115. EXTENDED_DATE_AND_TIME

RTC_CMD_STATUS

Este enumerador é utilizado para retornar o tipo de erro na configuração ou leitura do RTC:

Enumerador	Valor	Descrição
RTC_CMD_STATUS	NO_ERROR (0)	Não há erro.
	UNKNOWN_COMMAND (1)	Comando desconhecido.
	DEVICE_BUSY (2)	Dispositivo está ocupado.
	DEVICE_ERROR (3)	Dispositivo com erro.
	ERROR_READING_OSF (4)	Erro na leitura do sinalizador de data e hora válidas.
	ERROR_READING_RTC (5)	Erro na leitura da data e hora.
	ERROR_WRITING_RTC (6)	Erro na escrita da data e hora.
	ERROR_UPDATING_ SYSTEM_TIME (7)	Erro na atualização de data e hora do sistema.
	INTERNAL_ERROR (8)	Erro interno.
	INVALID_TIME (9)	Data e hora inválidas.
	INPUT_OUT_OF_RANGE (10)	Fora do limite de data e hora válidas para o sistema.
	TIMEZONE_CFG_NOT_SUPPORTED (11)	Erro gerado quando as configurações de fuso horário não são suportadas pelo dispositivo.

Tabela 4-116. RTC_STATUS

TIME ZONE SETTINGS

Esta estrutura é utilizada para armazenar o valor do fuso horário nas requisições de leitura/configuração dos blocos funcionais do RTC e é descrita na:

Estrutura	Tipo	Variável	Descrição
TIMEZONE SETTINGS	INT	iHour	Hora do fuso horário configurado.
	INT	iMinutes	Minuto do fuso horário configurado.

Tabela 4-117. TIMEZONE SETTINGS

Nota:

Blocos funcionais de escrita e leitura de data e hora: Bibliotecas diferentes da LibPlcStandard, que contenham blocos funcionais ou funções que possam fazer acesso de leitura e escrita da data e hora no sistema, não são indicados. A biblioteca LibPlcStandard possui as interfaces adequadas para escrever e ler a data e hora do sistema e informar os diagnósticos corretos.

Memória de Arquivos de Usuário

A UCP da Série Hadron Xtorm possui uma área de memória destinada ao armazenamento de dados de uso geral, ou seja, o usuário poderá gravar diversos arquivos de qualquer formato na memória da UCP. Esta área de memória pode variar de acordo com o modelo de UCP utilizado. Consultar capítulo Características Gerais – Notas.

Para usar esta área, o usuário deverá acessar um projeto no software MasterTool Xtorm e clicar na Árvore de Dispositivos, localizada à esquerda do programa. Deverá dar dois cliques sobre o item Device e, após selecionar a UCP na aba Configurações de Comunicação que será aberta, selecionar a aba Arquivos e clicar em Atualizar, tanto na coluna de arquivos do computador (esquerda), como na coluna de arquivos da UCP selecionada (direita), conforme mostram as indicações da Figura 4-49.

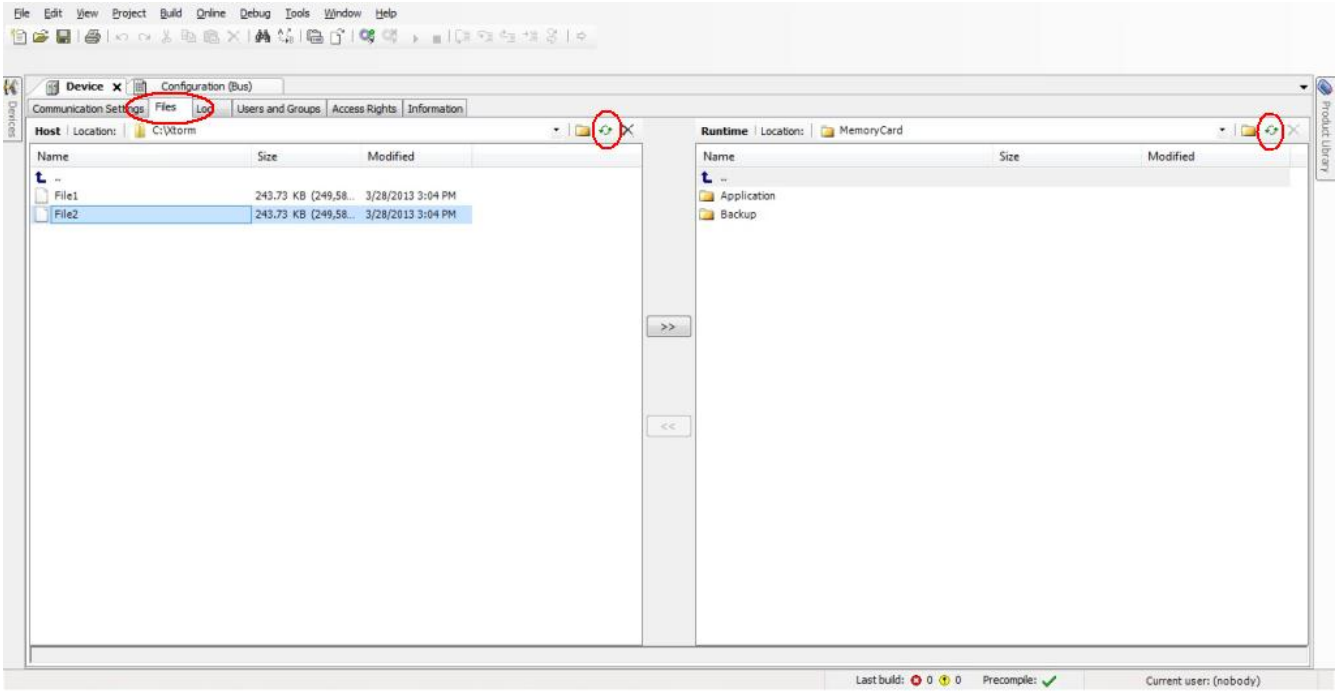


Figura 4-49. Acesso aos Arquivos de Usuário

Após atualizar a coluna de arquivos da UCP, será exibido o diretório raiz de arquivos armazenados na UCP e poderá ser selecionada a pasta para onde os arquivos serão transferidos. A pasta "InternalMemory" é uma pasta padrão, a ser utilizada para armazenar arquivos na memória interna da UCP (32 Mbytes), uma vez que não é possível transferir arquivos para o diretório raiz. Caso seja necessário, podem ser criadas outras pastas no diretório raiz ou subpastas dentro da pasta "InternalMemory". Já a pasta "MemoryCard" é o diretório onde o cartão de memória estará montado, caso o mesmo esteja inserido na UCP. Arquivos transferidos para a pasta "MemoryCard" estarão sendo transferidos diretamente para dentro do cartão de memória.

ATENÇÃO: No caso em que o cartão de memória seja inserido após a inicialização da UCP, será requisitado um usuário e senha para realizar o acesso e/ou transferências de arquivos do MasterTool Xtorm para o cartão de memória ou vice-versa. O usuário padrão com privilégios para acesso à UCP é "Owner" e a senha padrão desse usuário é "Owner".

Para realizar a transferência de algum arquivo do microcomputador para a UCP, basta selecionar o arquivo desejado na coluna da esquerda e pressionar o botão ">>", localizado no centro da tela, conforme Figura 4-50. O tempo de transferência irá variar de acordo com o tamanho do arquivo e com o tempo de ciclo (execução) da aplicação atual da UCP, podendo levar vários minutos.

O usuário não precisa estar em Modo Run ou conectado à UCP para realizar as transferências, pois ela possui a capacidade de se conectar automaticamente quando o usuário realizar a transferência.

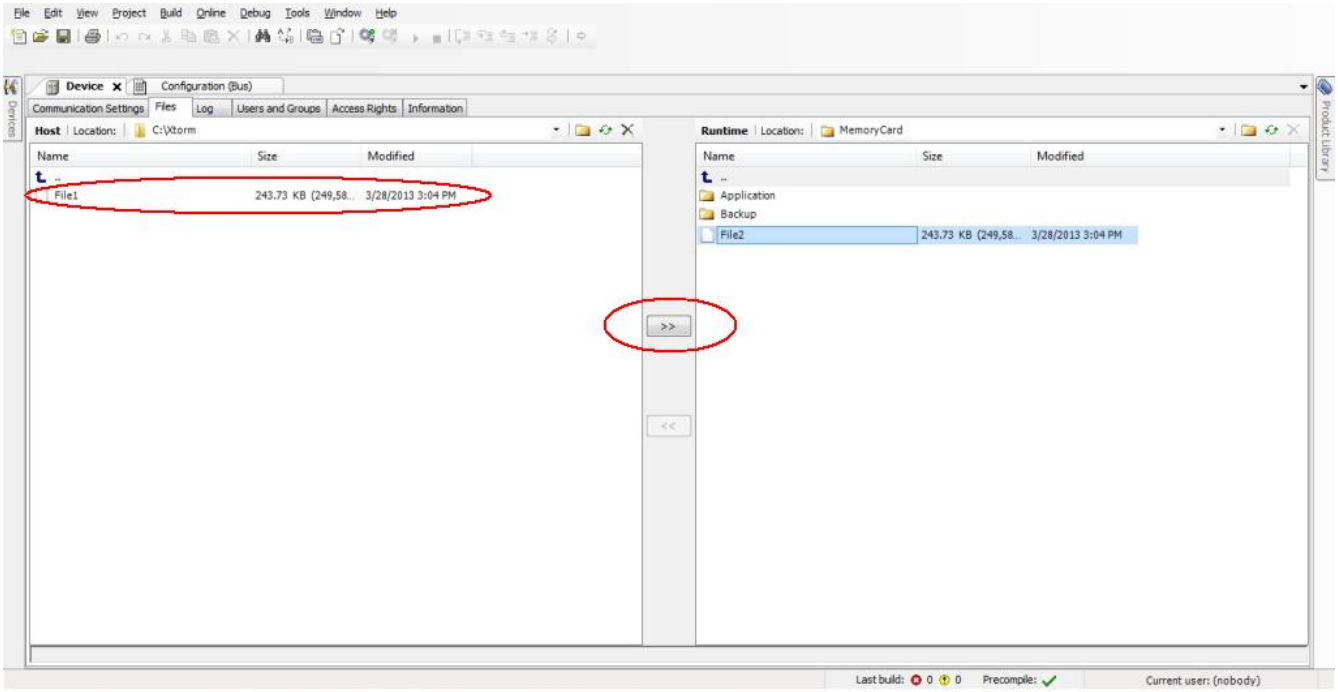


Figura 4-50. Transferindo Arquivos

ATENÇÃO: Os arquivos contidos dentro da pasta de um projeto criado pela ferramenta MasterTool Xtorm possuem nomes especiais reservados pelo sistema, desta forma não podem ser transferidos através da aba Arquivos. Caso o usuário deseje transferir um projeto para a memória de usuário, será necessário compactar a pasta e então transferir o arquivo compactado (*.zip por exemplo).

Caso seja necessário transferir documentos da UCP para o microcomputador em que está instalado o software MasterTool Xtorm o usuário deve realizar um procedimento muito semelhante ao anterior, ou seja, selecionando o arquivo na coluna da direita e pressionar o botão "<<", localizado no centro da tela.

Além disso, o usuário possui algumas opções de operação da área de armazenamento de arquivos, são elas:

Novo diretório : permite a criação de uma nova pasta na área de memória de usuário.

Excluir item : permite a exclusão de arquivos nos diretórios da área de memória de usuário.

Atualizar : permite atualizar, na tela do MasterTool Xtorm, os arquivos presentes na memória de usuário e no microcomputador.

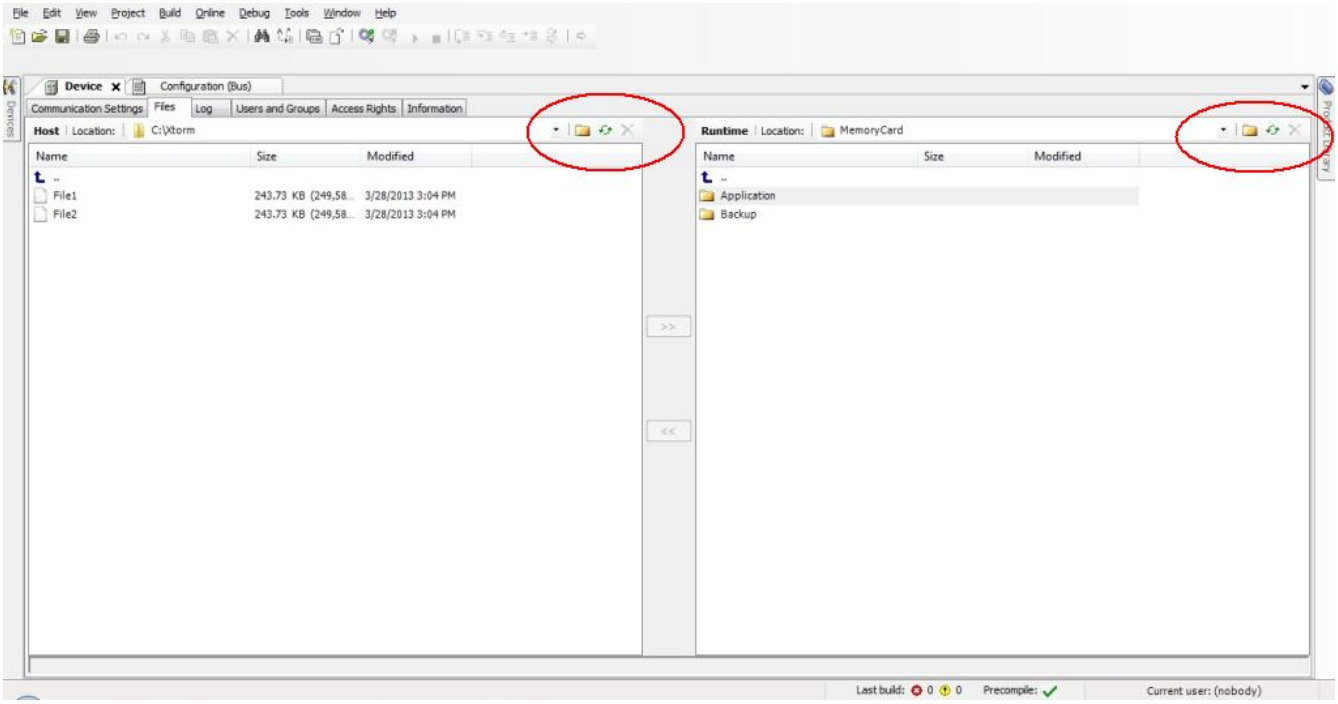


Figura 4-51. Opções de Utilização

ATENÇÃO: Para uma UCP em Modo Stop ou sem nenhuma aplicação, a taxa de transferência para a memória interna é de aproximadamente 150 Kbytes/s.

Cartão de Memória

Entre outras memórias, a UCP da Série Hadron Xtorm possibilita ao usuário a utilização de um cartão de memória, definido conforme as características do capítulo Descrição Técnica – **Interface do Cartão de Memória**, o qual serve para armazenar, entre outros arquivos, o projeto e a aplicação que está na memória interna da UCP.

Quando o cartão for inserido na UCP e estiver com sistema de arquivos diferente de FAT32, ela automaticamente identifica e pergunta ao usuário se ele deseja formatar. Em caso negativo, ele não poderá utilizar o cartão (o cartão não será montado, aparecerá uma mensagem dizendo que o formato não foi reconhecido, e o visor não indicará a presença do cartão). Caso seja selecionada a opção de formatação, a UCP irá levar alguns minutos, dependendo do tempo de ciclo (execução) da aplicação rodando na UCP, para executar a operação.

ATENÇÃO: Recomenda-se realizar o procedimento de formatação do cartão de memória diretamente na UCP Hadron Xtorm para evitar possíveis problemas de utilização, aumento do tempo de montagem ou até mesmo funcionamento incorreto. Não é recomendado remover o cartão de memória ou desernegizar a UCP durante a formatação ou durante a transferência de arquivos, pois pode causar a perda de dados bem como danos irreversíveis ao cartão.

Durante a configuração do projeto, dentro do software MasterTool Xtorm, o usuário habilita a opção de cópia do projeto da UCP para o cartão de memória ou a cópia do projeto que está no cartão para a UCP. Nessa mesma tela o usuário poderá configurar senhas de acesso, controlando o uso dessas informações. Informações sobre a tabela, consultar a seção Configuração da UCP - Cartão de Memória.

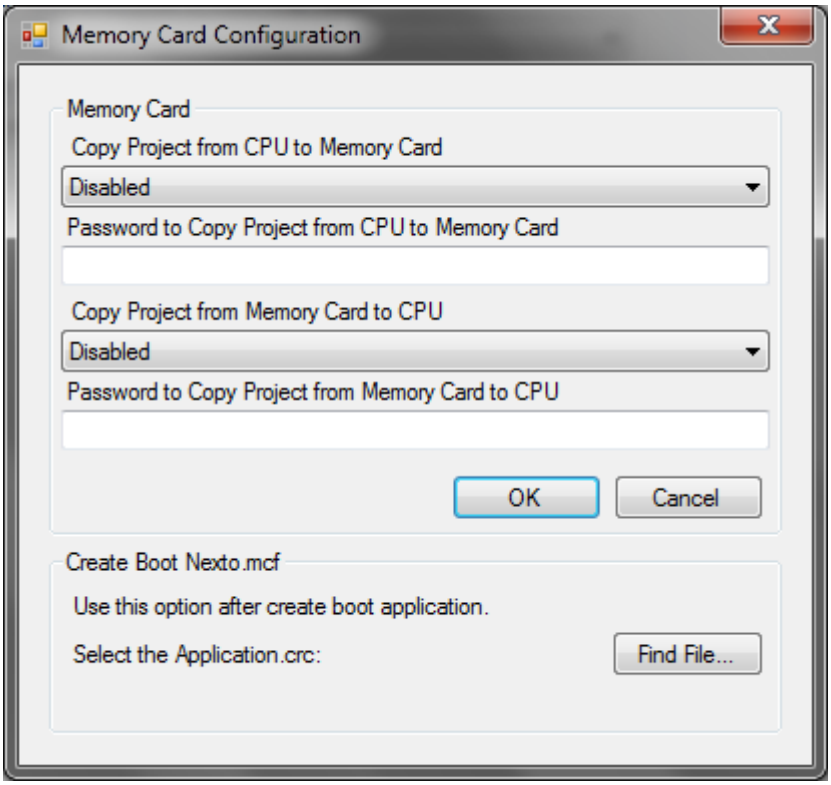


Figura 4-52. Configurações do Cartão de Memória

Quando uma senha é configurada para o cartão de memória no MasterTool, é necessário executar os passos a seguir para que, ao enviar o projeto, o arquivo criptografado que é gerado pelo MasterTool tenha a senha incluída no seu conteúdo e também seja enviado.

Primeiramente configurar a(s) senha(s) e clicar no botão “OK”. Neste momento as senhas foram registradas e o próximo passo é executar no menu Comunicação o comando “Criar Aplicação de Inicialização”, lembrando que não se pode estar logado na UCP para realizar esse procedimento. Após executar esse comando, três arquivos são criados. Um com a extensão “app” e outro com a extensão “crc”, a pasta onde o projeto está salvo também criará um arquivo com extensão “.ncf” necessário para a transferência do projeto no cartão para a UCP. Para concluir a operação de configuração da(s) senha(s) é necessário clicar novamente no botão “Cartão de Memória”, que está na configuração dos Parâmetros Gerais da UCP e então localizar o arquivo com extensão “crc” gerado no passo anterior, utilizando o botão “Localizar Arquivo...”. Executados esses passos, o MasterTool Xtorm irá enviar todos os arquivos necessários para realizar as operações de envio e recebimento de projetos via cartão de memória.

Caso o cartão esteja montado, a senha será gravada no mesmo. Caso contrário, a senha configurada no MasterTool será solicitada se o usuário tentar transferir o projeto da UCP para o cartão.

Para realizar o envio da UCP para o cartão de memória ou vice-versa, o usuário, além de habilitar no software MasterTool Xtorm e colocar a senha, terá que acessar o Menu “Cartão de Memória” na UCP, utilizando o botão de diagnósticos, e selecionar a opção de transferência desejada. Após, será solicitada a senha, caso o usuário tenha configurado durante a configuração da aplicação. Então, com um pressionamento curto no botão de diagnósticos os dígitos são incrementados e com um pressionamento longo são confirmados. No sexto dígito confirmado, a UCP irá consistir a senha e iniciará o processo. Quando as senhas, tanto da aplicação que está na UCP quanto da aplicação que está no cartão de memória, forem iguais, não é requisitado a inserção das senhas no menu da UCP para realizar as transferências das aplicações. Maiores informações sobre a utilização do botão de diagnósticos, consultar capítulo Manutenção – One Touch Diag.

Para remover o cartão de memória, basta acessar o Menu “Cartão de Memória” na UCP, utilizando o botão de diagnósticos, e selecionar a opção de remoção do cartão de memória, e aguardar até que o ícone do cartão desapareça da tela de status do visor gráfico.

ATENÇÃO:
Caso o cartão de memória seja removido sem ser desmontado via menu da UCP, durante a transferência de arquivos ou formatação, este processo pode acarretar na perda de dados no cartão, bem como corromper os arquivos nele contido. Este processo pode implicar na necessidade de uma nova formatação do cartão quando inserido novamente à UCP.

ATENÇÃO:
Caso exista algum arquivo na raiz do cartão de memória nomeado “Application” ou “Backup”, o mesmo será excluído para a criação das pastas de mesmo nome, utilizadas pela UCP para armazenamento do projeto aplicativo e do Project archive. Pastas com estes nomes não serão sobrescritas.

Acesso no MasterTool

O acesso à memória do cartão está vinculado à mesma tela de memória de usuário no software MasterTool Xtorm, sendo ele montado na pasta “MemoryCard”. Os diretórios “Application” e “Backup" são criados dentro do cartão de memória toda a vez que ele for inserido na UCP. Caso eles já existam, o sistema irá reconhecer e não sobrescreverá as pastas.

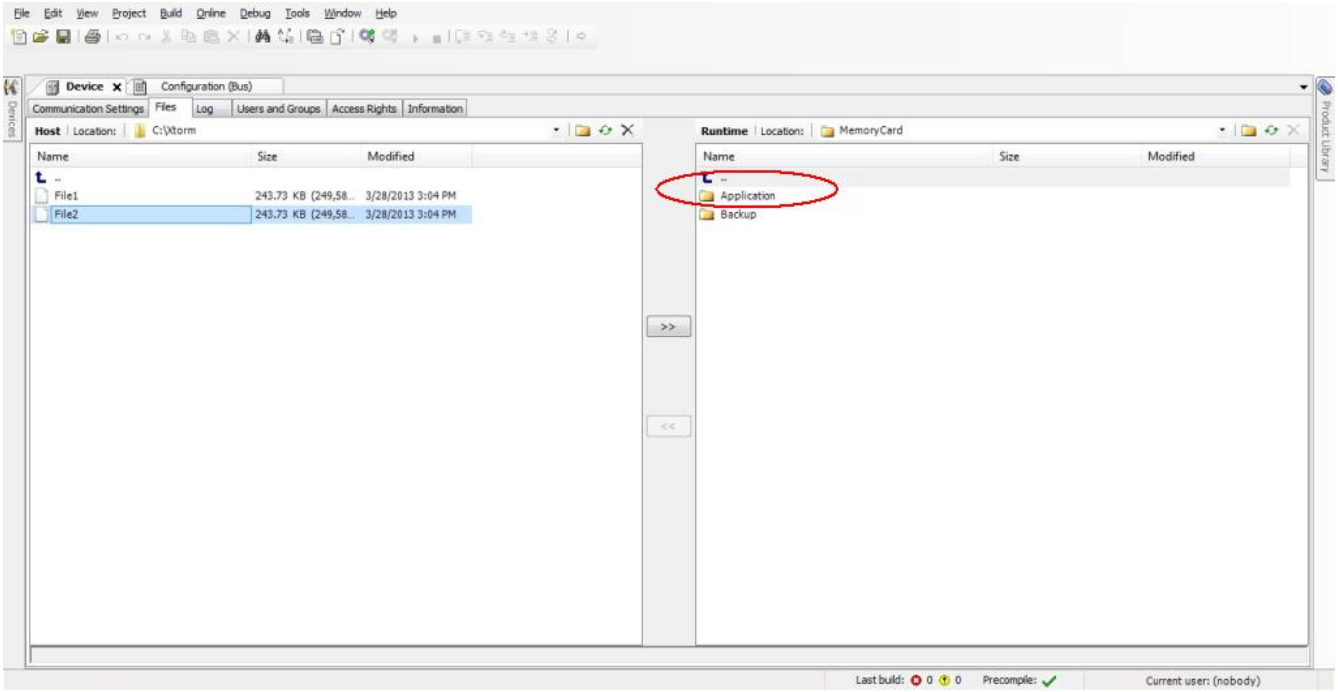


Figura 4-53. Diretório Raiz com Cartão de Memória Inserido

A transferência de arquivos ocorre de maneira semelhante à utilização da memória de usuário (Memória de Arquivos de Usuário), basta acessar o diretório “MemoryCard” e enviar os arquivos, conforme mostra a Figura 4-54.

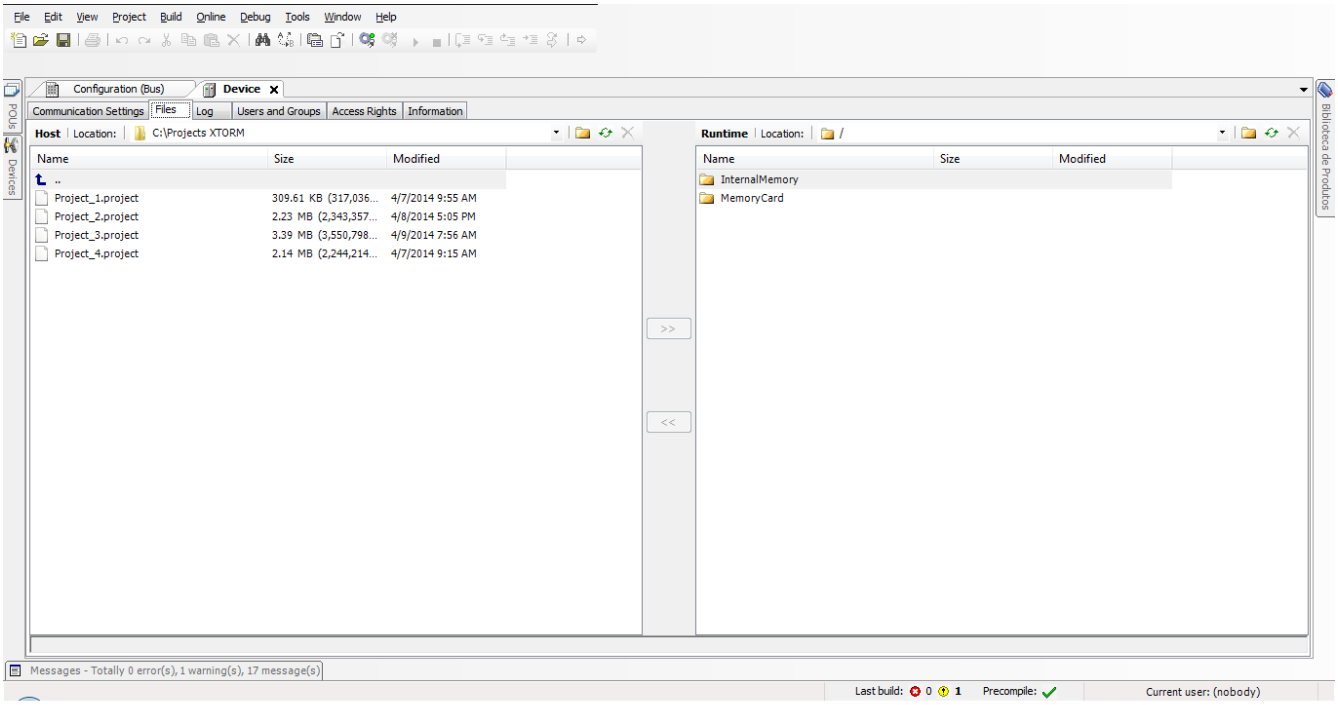


Figura 4-54. Arquivos Salvos no Cartão de Memória

Dentro do diretório do cartão de memória, além dos arquivos que estiverem armazenados no cartão, estarão as pastas “Application” e “Backup”. Nestas pastas ficarão salvos o aplicativo e o projeto atual caso o usuário escolha transferi-los ou fazer um backup dos mesmos através do menu da UCP.

ATENÇÃO: O tempo de transferência de arquivos depende da diferença do tempo de intervalo menos o tempo de execução médio da(s) tarefa(s) em execução (tempo disponível até o próximo ciclo da tarefa), isto é, quanto maior for essa diferença para cada tarefa em uma aplicação, mais rápida deverá ser a transferência de um dado a partir do cartão de memória para a UCP/MasterTool Xtorm ou vice-versa. A transferência de arquivos para o cartão de memória será mais lenta que a transferência para a memória interna da UCP. Para uma UCP em Modo Stop ou sem nenhuma aplicação, a taxa de transferência se aproxima de 100 Kbytes/s.

ATENÇÃO: O Mastertool Xtorm possui alguns nomes de arquivos reservados que não podem ser transferidos para o cartão de memória, são eles: Application.app, Application.crc, Archive.prj, Stdlogger.csv.

Menu Informativo e de Configuração da UCP

O acesso ao Menu Informativo e de Configuração da UCP Hadron Xtorm, assim como o acesso detalhado aos diagnósticos, estão disponíveis através de níveis, sendo que para acessar as informações do menu, trocar de nível e modificar alguma configuração, basta dar um pressionamento longo no botão de diagnóstico e, para navegar pelos itens de mesmo nível, basta dar um pressionamento curto no botão de diagnóstico. Consultar o capítulo Manutenção – One Touch Diag para verificar o funcionamento e a diferença entre tipos de pressionamento no botão de diagnósticos.

A Tabela 4-118 mostra os níveis do menu e o tipo de cada tela disponível na UCP, ou seja, se ela é de caráter informativo, configurável ou se retorna um nível.

Nível 1	Nível 2	Nível 3	Tipo
HARDWARE	TEMPERATURA	-	Informativo
	CONTRASTE	NIVEL CONTRASTE	Configurável
	DATA E HORA	-	Informativo
	VOLTAR	-	Retorna Nível
IDIOMAS	ENGLISH	>ENGLISH	Configurável
	PORTUGUES	>PORTUGUES	Configurável
	ESPANOL	>ESPANOL	Configurável
	VOLTAR	-	Retorna Nível
REDE	ENDERECO IP NET 1	-	Informativo
	MASCARA NET 1		Informativo
	ENDERECO IP NET 2		Informativo
	MASCARA NET 2		Informativo
	ENDERECO IP NET 3		Informativo
	MASCARA NET 3		Informativo
	ENDERECO IP NET 4		Informativo
	MASCARA NET 4		Informativo
	ENDERECO IP NET 5		Informativo
	MASCARA NET 5		Informativo
	ENDEREÇO IP NET 6		Informativo
	MASCARA NET 6		Informativo
	VOLTAR		Retorna Nível
SOFTWARE	FIRMWARE	-	Informativo
	BOOTLOADER		Informativo
	PROC. AUX.		Informativo
	VOLTAR		Retorna Nível
REDUNDANCIA	TROCAR PARA ATIVO ?	-	Informativo
	TROCAR PARA RESERVA?		Informativo
	VOLTAR		Retorna Nível
CARTAO DE MEM.	DESMONTAR		Configurável
	UCP > CARTAO	SENHA UCP	Configurável
	BACKUP		Configurável
	FORMATAR	CONFIRMA ?	Configurável
	CARTAO > UCP	SENHA CARTAO	Configurável
	VOLTAR	-	Retorna Nível
VOLTAR	-	-	Retorna Nível

Tabela 4-118. Níveis do Menu da UCP

Notas:

Cartão de Memória: O cartão de memória somente estará disponível no menu caso o mesmo esteja conectado na UCP Hadron Xtorm.

Senha: A senha para acesso aos dados do cartão de memória somente será necessária caso ela seja configurada no software MasterTool Xtorm. Não é possível editar a senha via menu. A senha precisa ter ao menos 6 caracteres.

Redundância: O menu "Redundancia" somente estará disponível caso a UCP HX3040 esteja identificada como Redundante .

Conforme apresentado pela Tabela 4-118, entre as opções disponíveis para visualização e alteração, encontram-se os principais dados necessários ao usuário, como:

- Informações sobre os recursos de hardware:
  - TEMPERATURA – Temperatura interna da UCP (Ex.: 36 C 97 F)
  - CONTRASTE – Ajuste do contraste do visor frontal da UCP
  - DATA E HORA – Data e hora configuradas na UCP (Ex.: 2001.01.31 00:00)
- Alteração do idioma do menu da UCP:
  - PORTUGUES – Altera o idioma para Português
  - ENGLISH – Altera o idioma para Inglês
  - ESPANOL – Altera o idioma para Espanhol
- Visualização de informações sobre a rede configurada no dispositivo:
  - ENDERECO IP NET 1 – Endereço IP (Ex.: 192.168.0.1)
  - MASCARA NET 1 – Máscara de subrede (Ex.: 255.255.255.0)
  - ENDERECO IP NET 2 – Endereço IP (Ex.: 192.168.1.1)
  - MASCARA NET 2 – Máscara de subrede (Ex.: 255.255.255.0)
  - NET 3: DESAB. (Caso não esteja configurada no projeto de usuário)
  - NET 4: DESAB. (Caso não esteja configurada no projeto de usuário)
  - NET 5: DESAB. (Caso não esteja configurada no projeto de usuário)
  - NET 6: DESAB. (Caso não esteja configurada no projeto de usuário)
- Informações sobre as versões de software:
  - FIRMWARE – Versão de software da UCP (Ex.: 1.0.0.0)
  - BOOTLOADER – Versão do bootloader da UCP (Ex.: 1.0.0.0)
  - PROC. AUX. – Versão do processador auxiliar da UCP (Ex.: 1.0.0.0)
- Acesso ao menu de opções de redundância da UCP:
  - TROCAR PARA ATIVO?
  - TROCAR PARA RESERVA:
- Acesso aos dados do Cartão de Memória:
  - CARTAO>UCP – Transferência do projeto do cartão de memória para a UCP
  - UCP>CARTAO – Transferência do projeto da UCP para o cartão de memória
  - FORMATAR – Formata o cartão para o sistema de arquivos FAT32
  - DESMONTAR – Habilita a remoção segura do cartão de memória da UCP
  - BACKUP – Faz um backup dos dados da UCP no cartão de memória

A Figura 4-55 descreve um exemplo de como operar o menu da UCP Hadron Xtorm, através do procedimento de ajuste do contraste a partir da tela de Status. Além de facilitar a configuração, é possível identificar todos os níveis de tela e o tipo de pressionamento para navegar entre as mesmas, sendo que para modificar os outros parâmetros, como Idioma e inserir a(s) senha(s) no Cartão de Memória, basta seguir a mesma lógica de acesso. O pressionamento curto mostra que o contraste está sendo incrementado (mais claro), sendo que no próximo pressionamento após o seu valor máximo, ele retorna ao valor mínimo (menos claro). O pressionamento longo mostra a confirmação do contraste desejado e o retorno ao nível anterior.

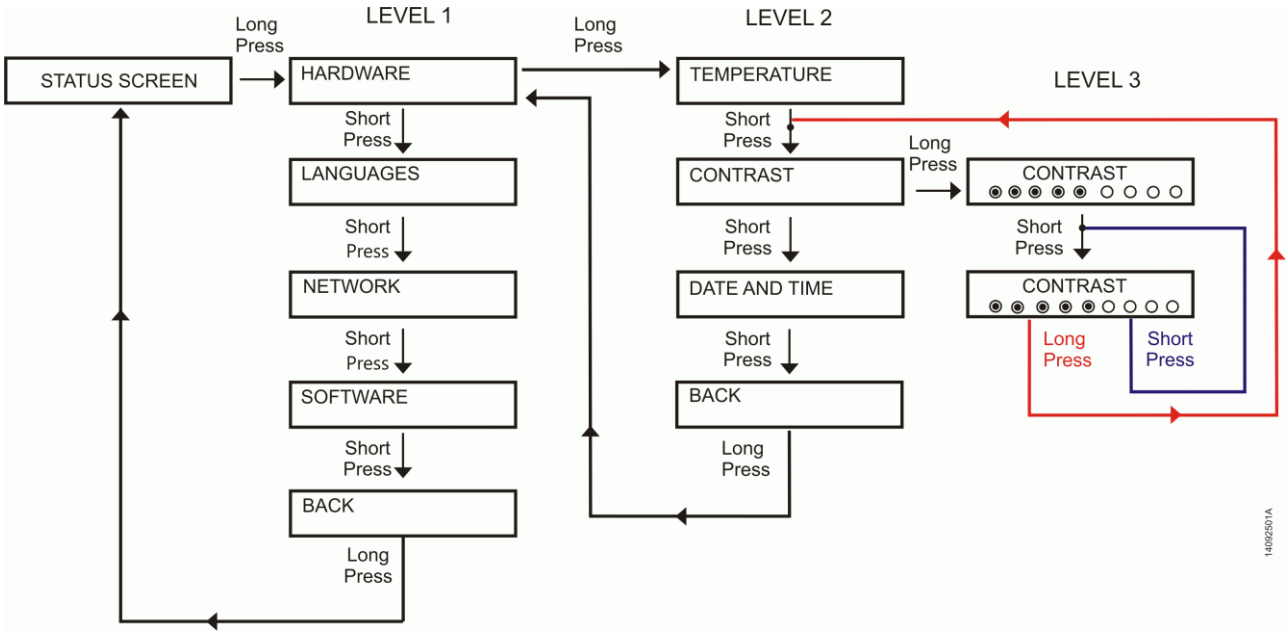


Figura 4-55. Ajuste do Contraste

Além do menu da UCP Hadron Xtorm ser encerrado através de um pressionamento longo no botão de diagnósticos na tela VOLTAR do nível 1, também existem outras condições de saída, as quais estão descritas abaixo:

- Pressionamento curto, em qualquer momento, nos outros módulos presentes no barramento, faz com que a UCP saia do menu e mostre os diagnósticos do módulo desejado
- Tempo de inatividade, em qualquer nível, superior a 5 s

Blocos Funcionais e Funções

Atualização de Entradas e Saídas

Funcionalidade utilizada para atualizar entradas e saídas no decorrer do aplicativo, não sendo necessário aguardar até que seja completado um ciclo.

Quando os blocos funcionais para atualizar as entradas e saídas não são utilizados, a atualização é realizada a cada ciclo da MainTask.

ATENÇÃO: Na inicialização de uma UCP da Série Hadron Xtorm, as entradas e saídas somente estarão atualizadas para leitura e preparadas para escrita quando a MainTask for executada. Todas as demais tarefas do sistema que executarem antes da MainTask estarão com as entradas e as saídas inválidas.

REFRESH_INPUT

Essa função é utilizada para atualizar as entradas do módulo especificado, sem aguardar o ciclo ser completado. É importante ressaltar que os filtros, configurados no MasterTool Xtorm, e o tempo de atualização das entradas do módulo, deverão ser considerados no tempo efetivo de atualização das entradas no aplicativo desenvolvido pelo usuário.

ATENÇÃO: a função de REFRESH_INPUT deve ser utilizada somente na tarefa MainTask. Caso seja necessária a atualização de E/S em outras tarefas, deve-se utilizar a opção de atualização de E/S por tarefa disponível na tela de configuração da UCP.

ATENÇÃO: a função de REFRESH_INPUT não suporta a atualização de entradas que tenham sido mapeadas para variáveis simbólicas. Para o correto funcionamento é necessário que a entrada esteja mapeada para uma variável dentro da memória de variáveis de entrada de representação direta (%I).

Figura 4-56. Função para Atualizar as Entradas

Parâmetros de entrada	Tipo	Descrição
byRackNumber	BYTE	Número do bastidor.
bySlotNumber	BYTE	Número da posição, na qual se encontra o módulo.

Tabela 4-119. Parâmetros de Entrada

Possíveis TYPE_RESULT:

- OK_SUCCESS: Execução com sucesso
- ERROR_MODULE_ABSENT: O módulo está ausente
- ERROR_MODULE_NOTCONFIGURED: O módulo não foi configurado
- ERROR_MODULE_COMMFAIL: Falha de comunicação com o módulo
- ERROR_FAILED: Esse erro é retornado caso a função seja chamada para um módulo que possua somente saídas, ou também caso a opção Sempre atualizar variáveis (localizada na tela de configuração do módulo, aba Mapeamento de E/S) não esteja marcada

Exemplo de utilização em Linguagem ST:

```
PROGRAM MainPrg
VAR
```

```
Info: TYPE_RESULT;
byRackNumber: BYTE;
bySlotNumber: BYTE;
END_VAR
//ENTRADAS:
byRackNumber := 0;
bySlotNumber := 10;
//FUNÇÃO:
Info:= REFRESH_INPUT ( byRackNumber, bySlotNumber); //Chamada da função.
//Variável 'Info' recebe possíveis erros da função.
```

REFRESH_OUTPUT

Essa função é utilizada para atualizar as saídas do módulo especificado, sem aguardar o ciclo ser completado. É importante ressaltar que o tempo de atualização das saídas do módulo deverá ser considerado no tempo efetivo de atualização das saídas no aplicativo desenvolvido pelo usuário.

ATENÇÃO: a função de REFRESH_OUTPUT deve ser utilizada somente na tarefa MainTask. Caso seja necessária a atualização de E/S em outras tarefas, deve-se utilizar a opção de atualização de E/S por tarefa disponível na tela de configuração da UCP.

ATENÇÃO: a função de REFRESH_OUTPUT não suporta a atualização de saídas que tiverem sido mapeadas para variáveis simbólicas. Para o correto funcionamento é necessário que a saída esteja mapeada para uma variável dentro da memória de variáveis de saída de representação direta (%Q).

Figura 4-57. Função para Atualizar as Saídas

Parâmetros de entrada	Tipo	Descrição
byRackNumber	BYTE	Número do bastidor.
bySlotNumber	BYTE	Número da posição, na qual se encontra o módulo.

Tabela 4-120. Parâmetros de Entrada

Possíveis TYPE_RESULT:

- OK_SUCCESS: Execução com sucesso
- ERROR_MODULE_ABSENT: O módulo está ausente
- ERROR_MODULE_NOTCONFIGURED: O módulo não foi configurado
- ERROR_MODULE_COMMFAIL: Falha de comunicação com o módulo
- ERROR_FAILED: Esse erro é retornado caso a função seja chamada para um módulo que possua somente entradas, ou também caso a opção Sempre atualizar variáveis (localizada na tela de configuração do módulo, aba Mapeamento de E/S) não esteja marcada

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
    Info: TYPE_RESULT;
    byRackNumber: BYTE;
    bySlotNumber: BYTE;
END_VAR
//ENTRADAS:
```

```
byRackNumber := 0;
bySlotNumber := 10;
//FUNÇÃO:
//Chamada da função.
Info := REFRESH_OUTPUT ( byRackNumber, bySlotNumber);
//Variável 'Info' recebe possíveis erros da função.
```

Timer Retentivo

O Timer Retentivo é um bloco funcional desenvolvido para aplicações como relógios de linha de produção, as quais necessitam armazenar o seu valor e reiniciar a contagem do mesmo ponto em caso de falha na alimentação. Os valores, guardados pelo bloco funcional somente serão zerados em caso de um Reset a Frio, Reset Origem ou o Download de uma nova aplicação, sendo que os contadores continuam em funcionamento mesmo que a aplicação esteja parada (Modo Stop).

ATENÇÃO: É importante destacar que, para o correto funcionamento dos blocos funcionais Timer Retentivo, as variáveis de controle devem ser declaradas como retentivas (VAR RETAIN).

Abaixo, são descritos os três tipos de blocos disponíveis na biblioteca LibPlcStandard do software MasterTool Xtorm (para realizar o procedimento de inserção de uma biblioteca, consultar capítulo Programação Inicial - Bibliotecas).

TOF_RET

O bloco funcional TOF_RET implementa um tempo de atraso para desabilitar uma saída. Quando a entrada IN tem seu estado alterado de verdadeiro (TRUE) para falso (FALSE), ou seja, uma borda de descida, o tempo especificado PT irá transcorrer até que a saída Q também seja falsa (FALSE). Quando a entrada IN tem nível lógico 1 (TRUE), a saída Q também permanecerá no mesmo estado (TRUE), mesmo que isso aconteça no meio de uma contagem. O tempo PT pode ser alterado durante a contagem, pois o bloco funcional assumirá o novo valor, desde que a contagem não tenha chegado ao final. A

Figura 4-58 representa o bloco TOF_RET e a Figura 4-59 mostra o comportamento gráfico do mesmo.

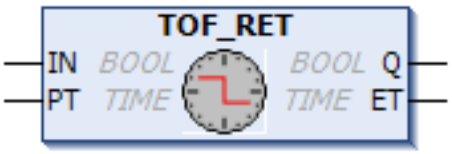


Figura 4-58. Bloco Funcional TOF_RET

Parâmetros de entrada	Tipo	Descrição
IN	BOOL	Essa variável, quando recebe uma borda de descida, habilita a contagem do bloco funcional.
PT	TIME	Essa variável especifica o limite de contagem do bloco funcional (tempo de atraso).

Tabela 4-121. Parâmetros de Entrada

Parâmetros de saída	Tipo	Descrição
Q	BOOL	Essa variável vai para FALSE assim que a variável PT (tempo de atraso) atinge o seu valor máximo.
ET	TIME	Essa variável exibe o valor atual do tempo de atraso.

Tabela 4-122. Parâmetros de Saída

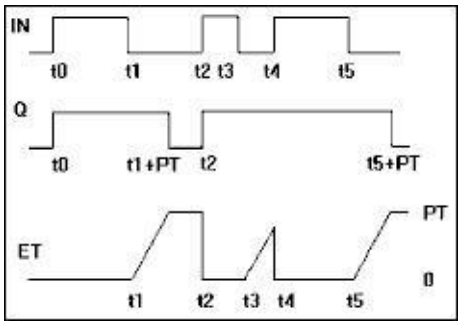


Figura 4-59. Comportamento Gráfico do Bloco Funcional TOF_RET

Exemplo de utilização em linguagem ST:

```
PROGRAM UserPrg
VAR RETAIN
   bStart : BOOL := TRUE;
   TOF_RET : TOF_RET;
END_VAR

// Quando bStart=FALSE inicia contagem
TOF_RET( IN := bStart,
PT := T#20S);

// Executa ações ao final da contagem
IF ( TOF_RET.Q = FALSE) THEN
   bStart := TRUE;
END_IF
```

TON_RET

O bloco funcional TON_RET implementa um tempo de atraso para habilitar uma saída. Quando a entrada IN tem seu estado alterado de falso (FALSE) para verdadeiro (TRUE), ou seja, uma borda de subida, o tempo especificado PT irá transcorrer até que a saída Q também seja verdadeira (TRUE). Quando a entrada IN tem nível lógico 0 (FALSE), a saída Q também permanecerá no mesmo estado (FALSE), mesmo que isso aconteça no meio de uma contagem. O tempo PT pode ser alterado durante a contagem, pois o bloco funcional assumirá o novo valor, desde que a contagem não tenha chegado ao final. A Figura 4-60 representa o bloco TON_RET e a Figura 4-61 mostra o comportamento gráfico do mesmo.

Figura 4-60. Bloco Funcional TON_RET

Parâmetros de entrada	Tipo	Descrição
IN	BOOL	Essa variável, quando recebe uma borda de subida, habilita a contagem do bloco funcional.
PT	TIME	Essa variável especifica o limite de contagem do bloco funcional (tempo de atraso).

Tabela 4-123. Parâmetros de Entrada

Parâmetros de saída	Tipo	Descrição
Q	BOOL	Essa variável vai para TRUE assim que a variável PT (tempo de atraso) atinge o seu valor máximo.
ET	TIME	Essa variável exibe o valor atual do tempo de atraso.

Tabela 4-124. Parâmetros de Saída

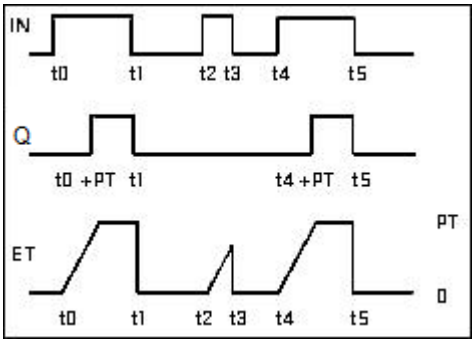


Figura 4-61. Comportamento Gráfico do Bloco Funcional TON_RET

Exemplo de utilização em linguagem ST:

```
PROGRAM UserPrg
VAR RETAIN
   bStart : BOOL;
   TON_RET : TON_RET;
END_VAR

// Quando bStart=TRUE inicia contagem
TON_RET( IN := bStart,
PT := T#20S);

// Executa ações ao final da contagem
IF ( TON_RET.Q = TRUE) THEN
   bStart := FALSE;
END_IF
```

TP_RET

O bloco funcional TP_RET trabalha como um trigger. O timer, que inicia quando a entrada IN tem seu estado alterado de falso (FALSE) para verdadeiro (TRUE), ou seja, uma borda de subida; é incrementado até que o limite de tempo PT seja atingido. Durante a contagem, a saída Q é verdadeira (TRUE), caso contrário ela é falsa (FALSE). O tempo PT pode ser alterado durante a contagem, pois o bloco assumirá o novo valor, desde que a contagem não tenha chegado ao final. A Figura 4-62 representa o bloco TP_RET e a Figura 4-63 mostra o comportamento gráfico do mesmo.

Figura 4-62. Bloco Funcional TP_RET

Parâmetros de entrada	Tipo	Descrição
IN	BOOL	Essa variável, quando recebe uma borda de subida, habilita a contagem do bloco funcional.
PT	TIME	Essa variável especifica o limite de contagem do bloco funcional (tempo de atraso).

Tabela 4-125. Parâmetros de Entrada

Parâmetros de saída	Tipo	Descrição
Q	BOOL	Essa variável é verdadeira durante a contagem. Caso contrário é falsa.
ET	TIME	Essa variável exibe o valor atual do tempo de atraso.

Tabela 4-126. Parâmetros de Saída

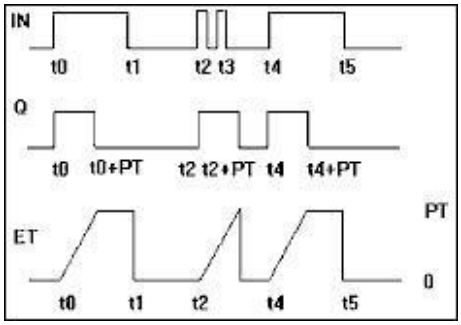


Figura 4-63. Comportamento Gráfico do Bloco Funcional TP_RET

Exemplo de utilização em linguagem ST:

```
PROGRAM UserPrg
VAR RETAIN
   bStart : BOOL;
   TP_RET : TP_RET;
END_VAR

// Configura TP_NR
TP_RET( IN := bStart,
PT := T#20S);

bStart := FALSE;

// Ações durante a contagem
IF ( TP_RET.Q = TRUE) THEN
// Executa enquanto o contador estiver ativado
ELSE
// Executa somente quando o contador estiver desativado
END_IF
```

Timer Não-Redundante

O timer não redundante é utilizado em aplicações para a UCP HX3040 redundante que necessitam de um timer no programa não redundante em uma arquitetura com redundância de UCP. Este timer não utiliza o timer IEC, portanto, não irá ser sincronizado na arquitetura de redundância de UCP caso a UCP reserva assumir o estado ativo e o ativo passar para reserva.

Abaixo, são descritos os três tipos de blocos já disponíveis na biblioteca LibPlcStandard do software MasterTool Xtorm (para realizar o procedimento de inserção de uma biblioteca, consultar capítulo Programação Inicial - Bibliotecas).

TOF_NR

O bloco funcional TOF_NR implementa um tempo de atraso para desabilitar uma saída e tem o funcionamento e configuração parecidos com o bloco funcional TOF_RET, se diferenciando apenas por não ser redundante e nem retentivo.

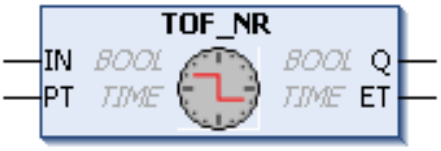


Figura 4-64. Bloco Funcional TOF_NR

Exemplo de utilização em linguagem ST:

```
PROGRAM NonSkippedProg
VAR
    bStart : BOOL := TRUE;
    TOF_NR : TOF_NR;
END_VAR

// Quando bStart=FALSE inicia contagem
TOF_NR( IN := bStart,
PT := T#20S);

// Executa ações ao final da contagem
IF ( TOF_NR.Q = FALSE) THEN
    bStart := TRUE;
END_IF
```

TON_NR

O bloco funcional TON_NR implementa um tempo de atraso para habilitar uma saída e tem o funcionamento e configuração parecidos com o bloco funcional TON_RET, se diferenciando apenas por não ser redundante e nem retentivo.

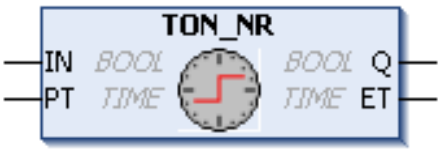


Figura 4-65. Bloco Funcional TON_NR

Exemplo de utilização em linguagem ST:

```
PROGRAM NonSkippedProg
VAR
    bStart : BOOL;
    TON_NR : TON_NR;
END_VAR

// Quando bStart=TRUE inicia contagem
TON_NR( IN := bStart,
PT := T#20S);

// Executa ações ao final da contagem
IF ( TON_NR.Q = TRUE) THEN
    bStart := FALSE;
END_IF
```

TP_NR

O bloco funcional TP_NR trabalha como um trigger e tem o funcionamento e configuração parecidos com o bloco funcional TP_RET, se diferenciando apenas por não ser redundante e nem retentivo.

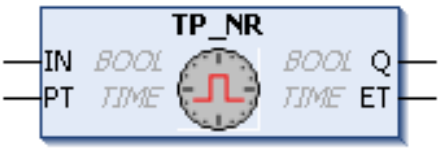


Figura 4-66. Bloco Funcional TP_NR

Exemplo de utilização em linguagem ST:

```
PROGRAM NonSkippedProg
VAR
   bStart : BOOL;
   TP_NR : TP_NR;
END_VAR

// Configura TP_NR
TP_NR( IN := bStart,
PT := T#20S);

bStart := FALSE;

// Ações durante a contagem
IF ( TP_NR.Q = TRUE) THEN
// Executa enquanto o contador estiver ativado
ELSE
// Executa somente quando o contador estiver desativado
END_IF
```

ClearRtuDiagnostic

O bloco funcional ClearRtuDiagnostic disponibilizado pela biblioteca LibRtuStandard pode ser utilizado quando for necessário para limpar os diagnósticos que indicam Perda de Retentividade, Overflow da Fila de Eventos e também pode resetar a flag de fila de eventos limpa. Conforme a Tabela 4-127 para cada código do ENUM relacionado à função retornará uma ação.

Nome	ENUM (BYTE)	Descrição
DIAG_RETENTIVITY_LOST	0	Limpa diagnóstico de Perda de Retentividade
DIAG_QUEUE_OVERFLOW	1	Limpa diagnóstico de Fila de Eventos cheia/estourada
DIAG_QUEUE_CLEARED	2	Limpa diagnóstico de fila de eventos limpa

Tabela 4-127. Descrição da função ClearRtuDiagnostic

Exemplo de utilização em linguagem ST, onde a chamada da função irá limpar o bit DG_HX3040.tSummarized.bRetentivityLost da estrutura de diagnósticos da UCP:

```
PROGRAM StartPrg
VAR
   Resultado : TYPE_RESULT;
END_VAR

Resultado := ClearRtuDiagnostic( eDiagnostic := 0);
```

ClearEventQueue

O bloco funcional ClearEventQueue disponibilizado pela biblioteca LibRtuStandard pode ser utilizado quando for necessário limpar a fila de eventos da UCP e de todos os drivers instanciados. Conforme a Tabela 4-128 o status de execução da função será mostrado na sua variável de saída.

Nome	Tipo	Descrição
ClearEventQueue	Type_Result	Variável que indica o resultado da execução do comando (sucesso ou falha)

Tabela 4-128. Descrição da função ClearEventQueue

Exemplo de utilização em linguagem ST, onde a chamada da função irá limpar todos os sub-diagnósticos contidos na árvore de diagnósticos T_DIAG_DNP_SERVER_1*:

```
PROGRAM UserPrg
VAR
   ClearEventQueueStatus : Type_Result;
END_VAR

ClearEventQueueStatus := ClearEventQueue();
```

Gerenciamento de Usuários e Direitos de Acesso

Fornecem funções para definir contas dos usuários e configurar os direitos de acesso ao projeto. Observe que o gerenciamento de usuários específico do dispositivo deve ser suportado para controlar os direitos de acesso no sistema de arquivos do CP e objetos durante a execução.

Os direitos para acessar os objetos do projeto via ações especificadas são atribuídos somente a grupos e, portanto, cada usuário deve pertencer a um grupo.

Gerenciamento de Usuários e Direitos de Acesso do Projeto

Gerenciamento de Usuários

A configuração dos usuários e grupos é feita no diálogo *Projeto* na janela *Configurações do Projeto*.

Os projetos contam, automaticamente, com um grupo padrão, chamado Everyone. Todos os usuários de outros grupos são membros deste grupo específico. Assim, cada conta de usuário é fornecida automaticamente com, no mínimo, as configurações padrão. O grupo *Everyone* não pode ser excluído, (somente renomeado) e os seus membros não podem ser removidos.

O projeto apresenta também o chamado grupo Owner que contém o usuário *Owner*. Usuários podem ser acrescentados ou removidos deste grupo, mas ao menos um usuário deve permanecer. Este grupo também não pode ser excluído e sempre tem todos os direitos de acesso. Tanto o grupo *Owner*, como o usuário *Owner* podem ser renomeados.

Ao iniciar o programador e um projeto, não há nenhum usuário conectado. Entretanto, o logon do usuário pode ser realizado através de uma conta definida com nome e senha e, assim, ele pode obter direitos de acesso específicos.

Observe que cada projeto tem o seu próprio gerenciamento de usuários, por exemplo, para ter direitos de acesso específicos em uma biblioteca, o usuário deve realizar o logon nesta biblioteca separadamente. Os usuários e grupos definidos em projetos diferentes não são idênticos, mesmo que tenham nomes iguais.

ATENÇÃO:
- As senhas dos usuários são armazenadas de forma irreversível. Caso uma senha seja perdida, a respectiva conta do usuário não mais poderá ser utilizada. Se a senha Owner for perdida, o projeto inteiro pode ser inutilizado.
- Por padrão, em novos projetos, a senha do usuário *Owner* é vazia.

Usuários

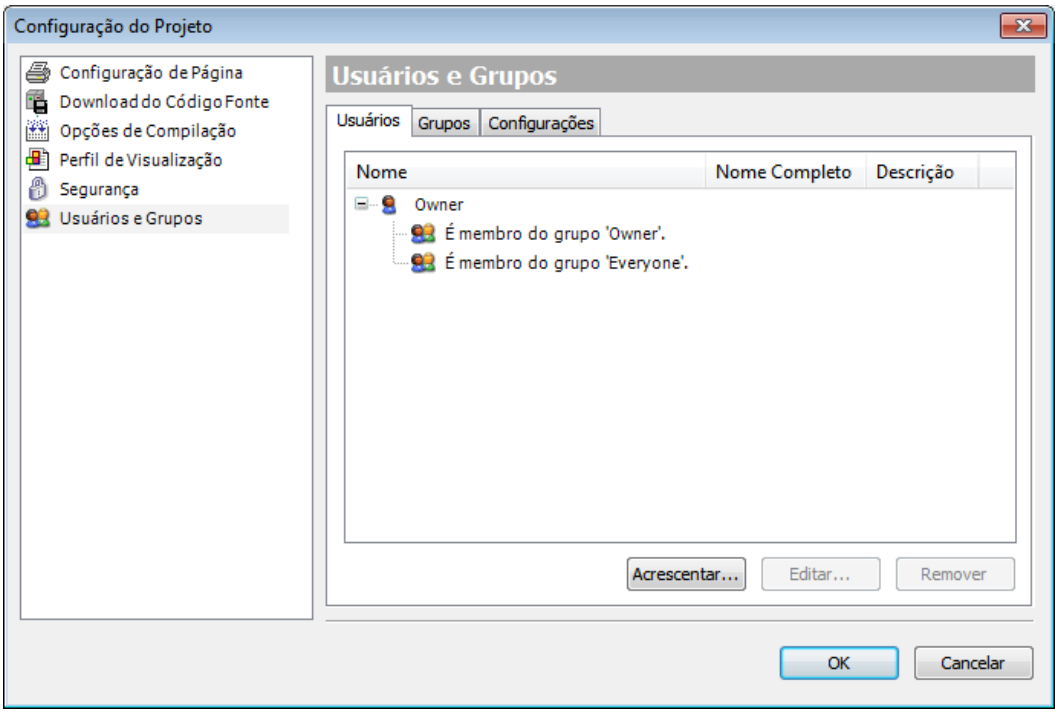


Figura 4-67. Configurações do Projeto, Diálogo Usuários

Os usuários atuais registrados são listados em uma estrutura de árvore. Através dos comandos *Acrescentar* ou *Editar* é possível exibir, além do Nome (logon), o Nome completo e a Descrição do usuário. As propriedades de cada usuário podem ser visualizadas ou não (ficam ocultas) através do sinal de mais e menos respectivamente. Cada usuário, por padrão, é membro do grupo “Everyone”.

Para definir uma nova conta de usuário, use o botão *Acrescentar* para abrir o diálogo *Acrescentar Usuário*.

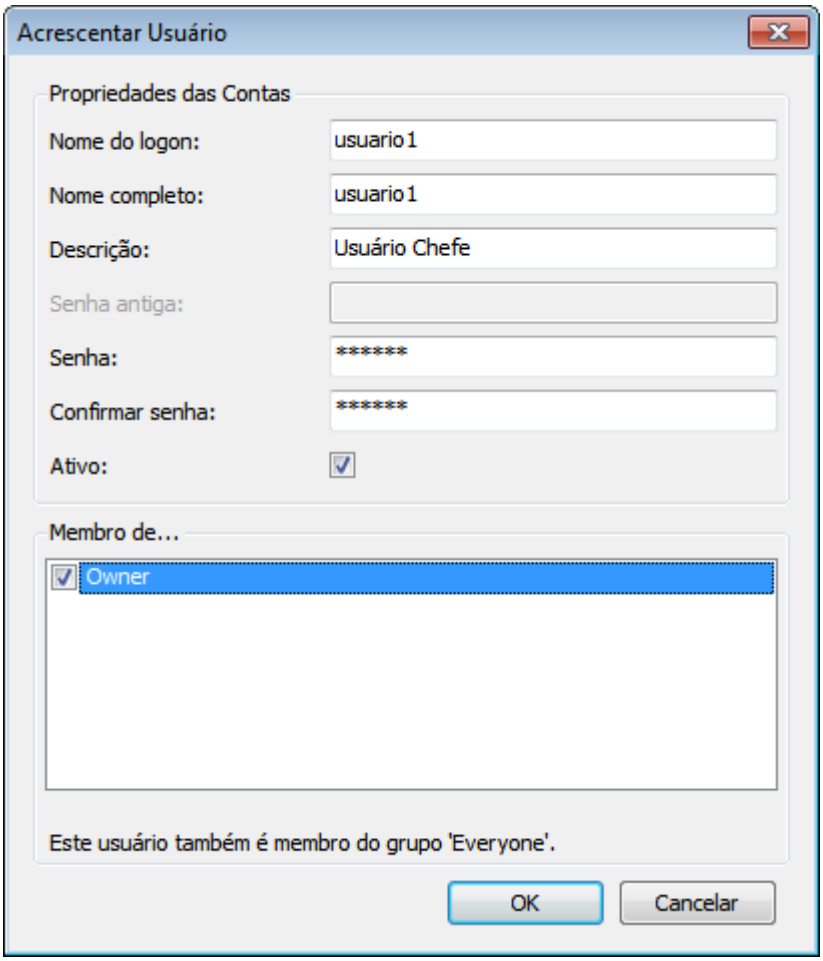


Figura 4-68. Acrescentar Usuário

Propriedades das contas, contém os seguintes campos:

- *Nome do logon*: nome de logon do novo usuário.
- *Nome completo*: nome completo do novo usuário. Serve somente como uma informação adicional.
- *Descrição*: descrição do novo usuário. Serve somente como uma informação adicional.
- *Senha antiga*: este campo é editável somente quando o diálogo é usado para modificar a conta de um usuário existente. Antes de alterar a senha, entretanto, é necessário digitar a senha atual válida.
- *Senha*: senha para o novo usuário. A digitação é representada por asteriscos (*).
- *Confirmar senha*: a senha digitada anteriormente deve ser repetida e, caso as duas senhas não coincidam, será mostrada uma mensagem de erro. Esta digitação também é representada por asteriscos (*).
- *Ativo*: esta opção ativa a conta do usuário, tornando-a válida. Quando a conta não é válida, o usuário não pode realizar o logon. A conta será automaticamente desativada quando forem feitas repetidas tentativas de logon com a senha incorreta.
- *Membro de...:* nesta lista são apresentados todos os grupos existentes, além do grupo "Everyone", ao qual pertence o novo usuário automaticamente. Selecionando os respectivos itens com (☑) se define a quais grupos o novo usuário deve pertencer.

Para configurar o novo usuário, feche o diálogo com *OK*. No caso de alguma incoerência (senha incorreta, ausência de nome de logon, usuário já existente), será apresentada uma mensagem de erro.

Para modificar uma conta de usuário existente, use o botão *Editar* e abra o diálogo correspondente. Estes campos são iguais aos do diálogo *Acrescentar Usuário*. Os campos da senha - por questões de

segurança - apresentarão 32 asteriscos (*). Após ter modificado os itens desejados, feche o diálogo com *OK* para aplicar as novas configurações.

Para excluir uma ou várias contas de usuários, selecione os respectivos usuários na lista apropriada e tecle *Remover*. Note que esta ação não exige confirmação. Não é possível excluir todos os usuários do grupo (no mínimo um deve permanecer). Caso o usuário tente fazer isto, aparecerá uma mensagem de erro.

Grupos

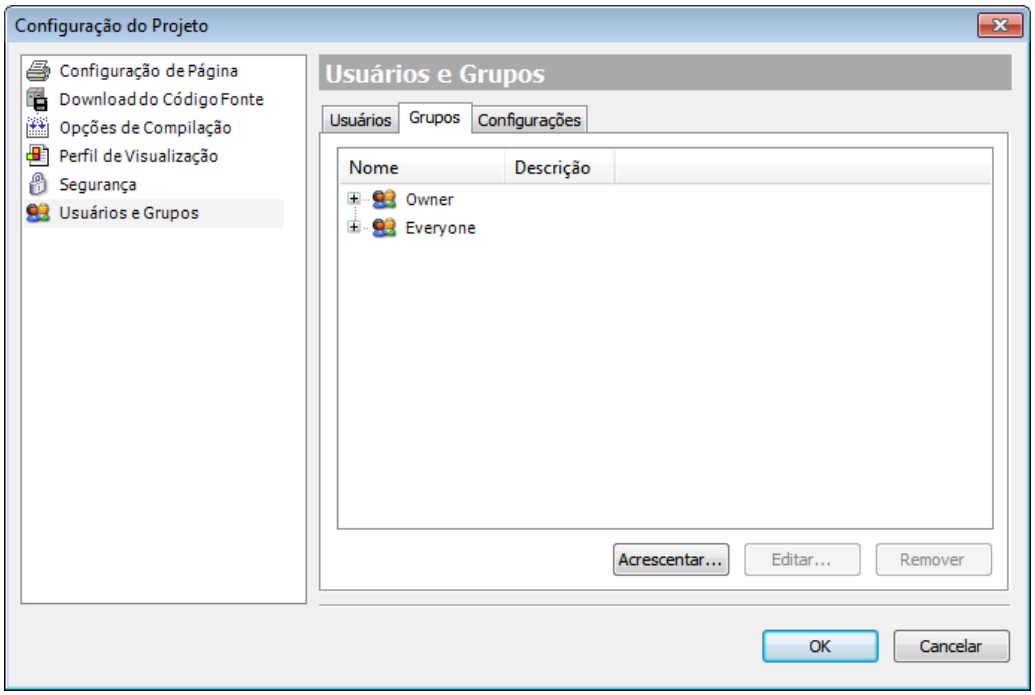


Figura 4-69. Configurações de Projeto, Diálogo Grupos

Os grupos atuais disponíveis são listados em uma estrutura de árvore. Os membros de cada grupo podem ser visualizados ou não através do sinal de mais e menos respectivamente. Lembre que o membro deve ser parte de um grupo.

Para acrescentar um novo grupo, utilize o botão *Acrescentar* e abra o diálogo correspondente.

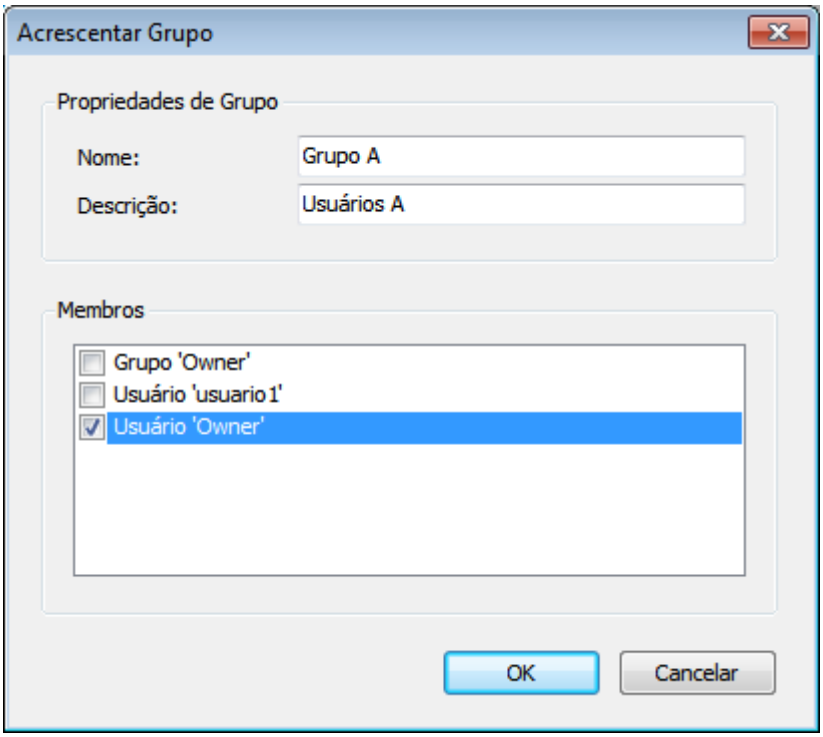


Figura 4-70. Acrescentar Grupo

Os seguintes campos devem ser preenchidos:

- *Nome*: nome do novo grupo.
- *Descrição*: descrição do novo grupo. Serve somente como uma informação adicional.
- *Membros*: nesta lista estão apresentados todos os usuários e grupos. Selecione aqueles que devem fazer parte do grupo atual com (☑).

Para configurar o novo grupo, feche o diálogo com *OK*. No caso de alguma incoerência (ausência de nome, grupo já existente, seleção de um grupo que provocaria um "ciclo de grupo"), será apresentada uma mensagem de erro.

Para modificar um grupo existente, use o botão *Editar* para abrir o diálogo *Editar Grupo*. Os campos são iguais aos do diálogo *Acrescentar Grupo* (Figura 4-70). Os campos da senha - por questões de segurança - apresentarão 32 asteriscos (*). Após ter modificado os itens desejados, feche o diálogo com *OK* para aplicar as novas configurações.

Para remover um ou vários grupos, selecione os respectivos grupos na árvore e tecle *Remover*. Note que esta ação não exige confirmação. Os membros dos grupos excluídos permanecerão inalterados. Não é possível excluir os grupos *Everyone* e/ou *Owner*. Caso o usuário tente fazer isto, aparecerá uma mensagem de erro.

Configurações

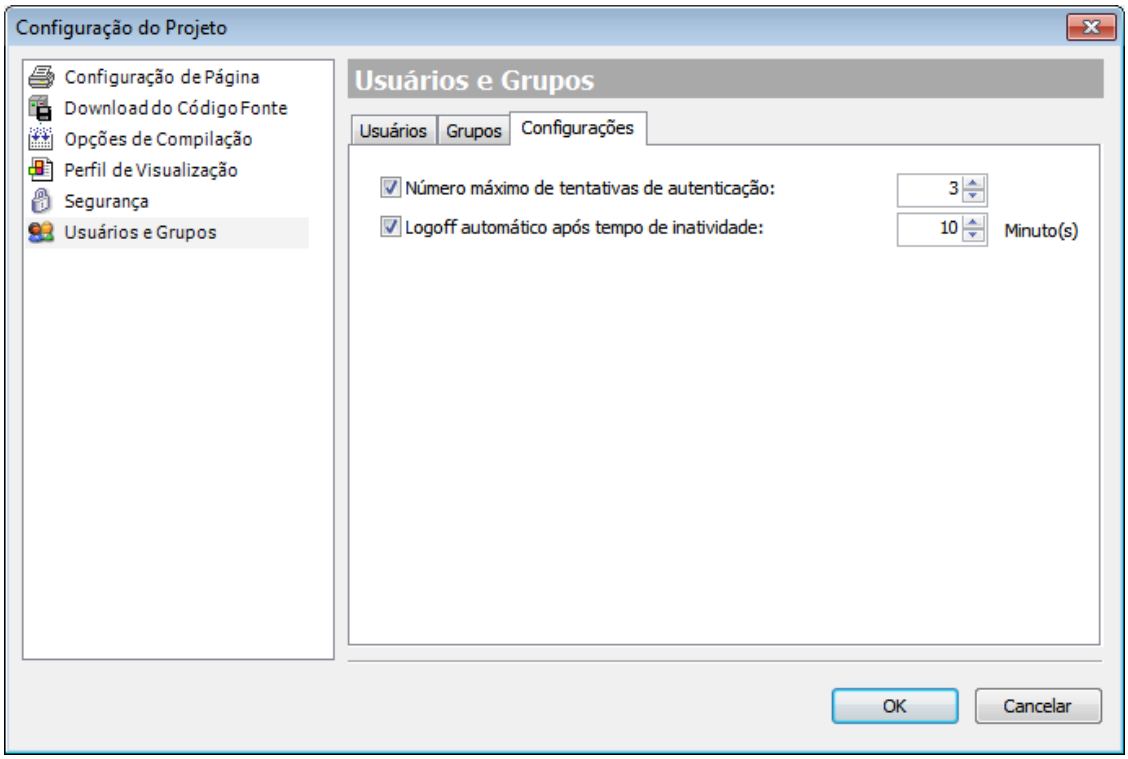


Figura 4-71. Configurações de Projeto, Diálogo Configuração de Usuários e Grupos

As seguintes configurações básicas referentes às contas do usuário podem ser realizadas:

- *Número máximo de tentativas de autenticação*: se esta opção está ativa, a conta do usuário se tornará inválida após o número especificado de tentativas de realizar o logon com a senha incorreta. Se a opção não estiver ativada, o usuário poderá realizar tantas tentativas quantas quiser. Padrão: opção ativada (3 tentativas). Valores permitidos: 1-10.
- *Logoff automático após tempo de inatividade*: se esta opção está ativa, a conexão da conta do usuário será perdida automaticamente após determinado tempo de inatividade (ausência de ações via mouse ou teclado). Padrão: opção ativada (10 minutos). Valores de tempo permitidos: 1-180 minutos.

Gerenciamento de Direitos de Acesso

O gerenciamento de usuários em um projeto somente é útil se combinado com o gerenciamento dos direitos de acesso.

Em um novo projeto, basicamente todos os direitos de acesso não são definidos automaticamente, mas configurados para um valor padrão, ou seja, normalmente os direitos estão “garantidos”.

Durante a execução do projeto, cada direito pode ser explicitamente garantido ou negado e configurado novamente para o padrão. O gerenciamento dos direitos de acesso é feito no diálogo *Permissões* ou - para os direitos de acesso aos objetos - no diálogo *Controle de Acesso* (que faz parte do diálogo *Propriedades do Objeto*).

Os direitos de acesso aos objetos são “herdados”. Se o objeto tem um objeto “principal”, os direitos de acesso deste tornam-se as configurações padrão do objeto secundário (exemplo: se uma ação é atribuída a um programa, ela é inserida na sua estrutura. Assim, o programa é o objeto “principal” da ação) . No que se refere aos direitos de acesso, normalmente as relações dos objetos principais - secundários correspondem às relações mostradas na árvore das POUs ou Dispositivos e são indicadas no diálogo *Permissões* através da sintaxe “<objeto principal>.<objeto secundário>”.
Exemplo: Ação ACT é atribuída ao objeto MainPrg (POU). Assim, na janela das POUs, ACT é

mostrada na árvore de objetos em MainPrg. No diálogo *Permissões*, ACT é representada por "MainPrg.ACT" indicando que MainPrg é o "principal" da ACT. Se o direito "modificar" fosse explicitamente negado para MainPrg e a um determinado grupo de usuários, o valor padrão deste direito para ACT também seria "negado" automaticamente.

Para acessar a tela *Permissões* deve-se clicar nessa opção no menu *Projeto* > *Gerenciamento de Usuário*. Será aberta a tela da Figura 4-72.

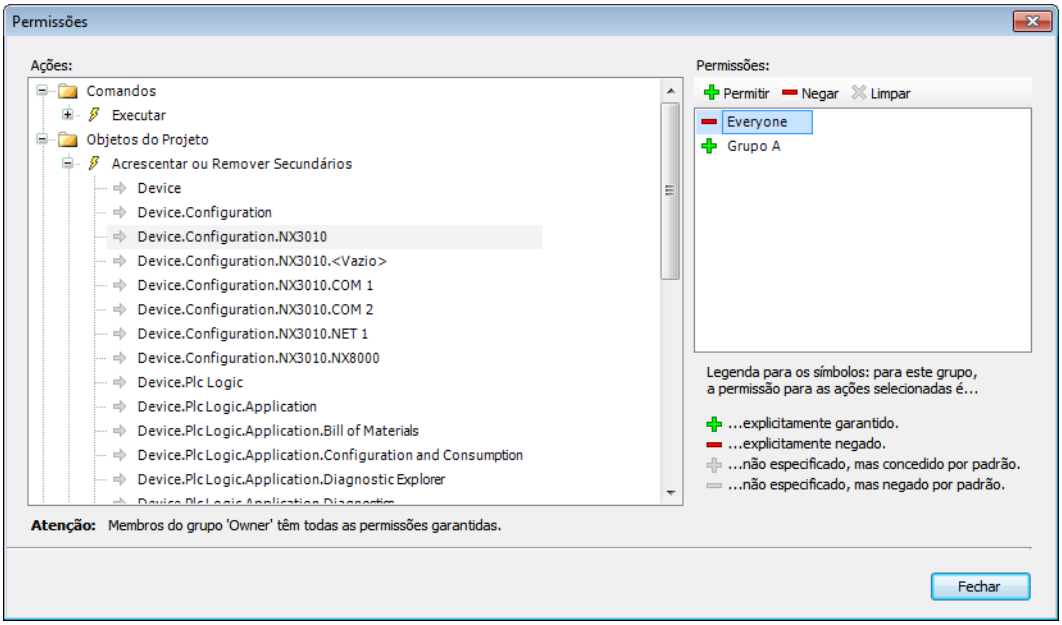


Figura 4-72. Permissões

A janela *Ações* exibe todas as possibilidades de direitos, ou seja, todas as ações que podem ser executadas nos objetos. A árvore é estruturada da seguinte forma:

- : No topo da árvore encontram-se os nomes de algumas categorias configuradas para estruturar visualmente os direitos de acesso. Estas categorias referem-se à execução de Comandos, à configuração de Contas de usuários e grupos, à criação de Tipos de objetos e também à visualização, edição, exclusão e tratamento de objetos secundários nos Objetos do projeto.
- : Em cada categoria encontram-se nós com as ações específicas que podem ser executadas em relação ao comando, conta do usuário, grupo, tipo de objeto ou objeto do projeto. Estes nós possuem também apenas função visual.

Possíveis ações:

Executar (execução de um comando do menu).
Criar (criação de um novo objeto no projeto atual).
Acrescentar ou excluir objetos secundários (de um objeto existente).
Modificar (edição de um objeto no editor).
Excluir (excluir ou recortar um objeto).
Visualizar (visualização de um objeto no editor).
: Cada item "ação" pode conter "dispositivos", ou seja, objetos (do projeto).

A janela Permissões fornece uma lista de todos os grupos de usuários disponíveis (exceto o grupo *Owner*) e uma barra de ferramentas para configurar os direitos de acesso ao grupo.

À esquerda de cada nome do grupo encontram-se ícones que indicam a permissão atual referente ao dispositivo selecionado na janela *Ações*:

- : a ação selecionada atualmente na janela Ações é garantida ao grupo.
- : a ação selecionada atualmente na janela Ações é negada ao grupo.

- : o direito de executar uma ou várias ações selecionadas na janela Ações não é garantido explicitamente, mas sim por padrão, devido ao referido direito ter sido garantido ao objeto "principal". Basicamente esta é a configuração padrão para todos os direitos que não tenham sido explicitamente configurados.
- : o direito de executar uma ou várias ações selecionadas na janela Ações não é negado explicitamente, mas sim por padrão, por exemplo, nos casos em que o referido direito tenha sido atribuído ao objeto principal.

Os ícones não serão exibidos caso sejam selecionadas várias ações sem as configurações únicas referentes ao grupo selecionado no momento.

Para configurar os direitos para um grupo selecione a ação e o grupo desejados nas janelas *Ações* e *Permissões* respectivamente. Após, use os botões apropriados na barra de ferramentas da janela *Permissões*:

- Permitir: Garantir explicitamente.
- Negar: Negar explicitamente.
- Limpar: Os direitos garantidos para a ação selecionada atualmente na janela Ações serão excluídos (retorno à configuração padrão).

Gerenciamento de Usuários e Direitos de Acesso da UCP

As UCPs HX3040 possuem um sistema de gerenciamento de permissões de usuário, que bloqueia ou permite certas ações para cada grupo de usuários na UCP. Para editar estes direitos na UCP, o usuário necessita acessar um projeto no software MasterTool Xtorm, não sendo necessário estar logado na UCP. Deverá então clicar na Árvore de Dispositivos, localizada à esquerda do programa, dar dois cliques no item *Device* e, após, selecionar a UCP na aba *Configurações de Comunicação* que será aberta. Apenas as abas *Usuários e Grupos* e *Direitos de Acesso* se relacionam com este tópico. A Figura 4-73 exemplifica os passos para acessar estas abas da UCP.

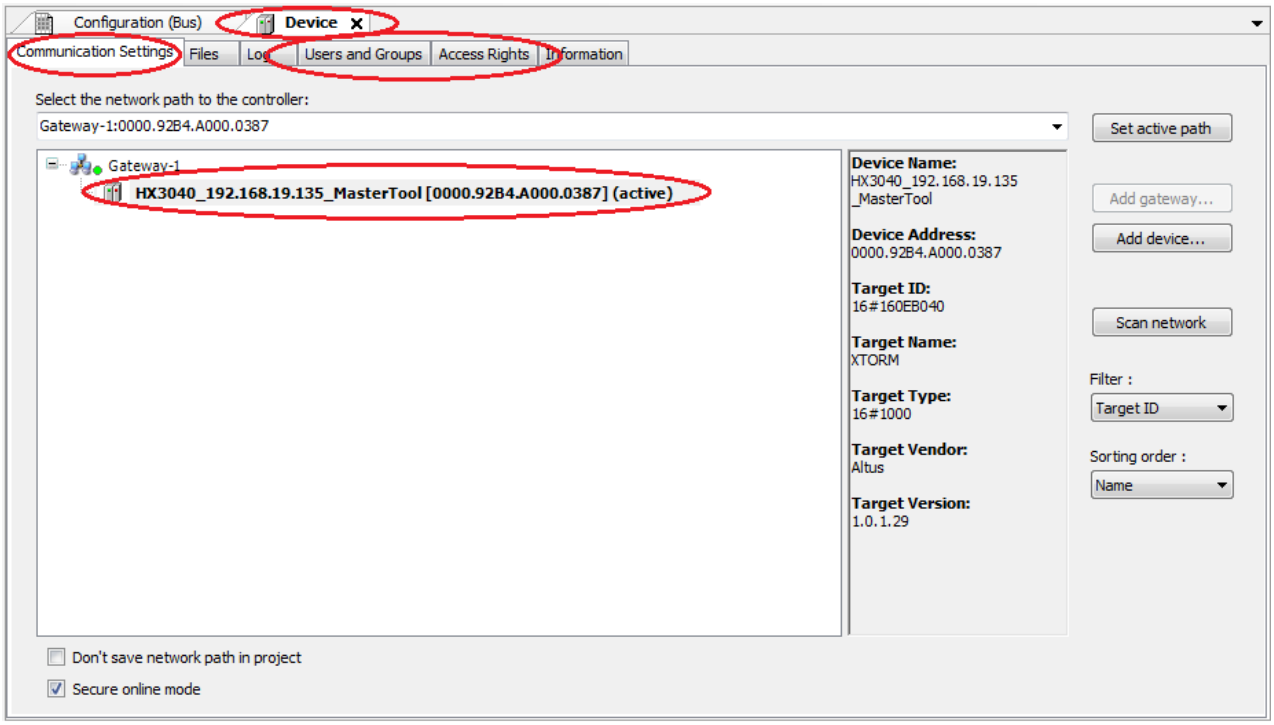


Figura 4-73. Acesso às Abas Usuários e Grupos e Direitos de Acesso

ATENÇÃO: - Caso o usuário esqueça a(s) senha(s) da(s) conta(s) com acesso à UCP, o único modo de recuperar este acesso será atualizando o firmware da mesma.

Usuários e Grupos

O diálogo *Usuários e Grupos* é fornecido em uma guia do diálogo *Device*. Ele permite configurar contas de usuários e grupos que, em conjunto com o gerenciamento dos direitos de acesso controlam o acesso aos objetos no CP no modo online.

Comum

Para que algumas funções de um controlador possam ser executadas apenas por usuários autorizados, utiliza-se o *Gerenciamento de Usuário Online*. Esta opção fornece a possibilidade de definir contas de usuários, atribuir direitos de acesso para grupos e forçar a autenticação do usuário no login.

O gerenciamento de usuários específico do dispositivo pode ser pré-definido pela descrição deste. Depende também da descrição do dispositivo quais definições podem ser editadas nos diálogos de configuração no programador.

Da mesma forma que no gerenciamento de usuários do projeto, os usuários devem ser membros dos grupos e somente grupos de usuários podem obter determinados direitos de acesso.

Usando o Diálogo de Configuração

Basicamente, o tratamento dos diálogos de gerenciamento de usuários online é similar ao do gerenciamento de usuários do projeto. Há a possibilidade de "importar" definições de contas de usuários a partir do gerenciamento de usuários do projeto.

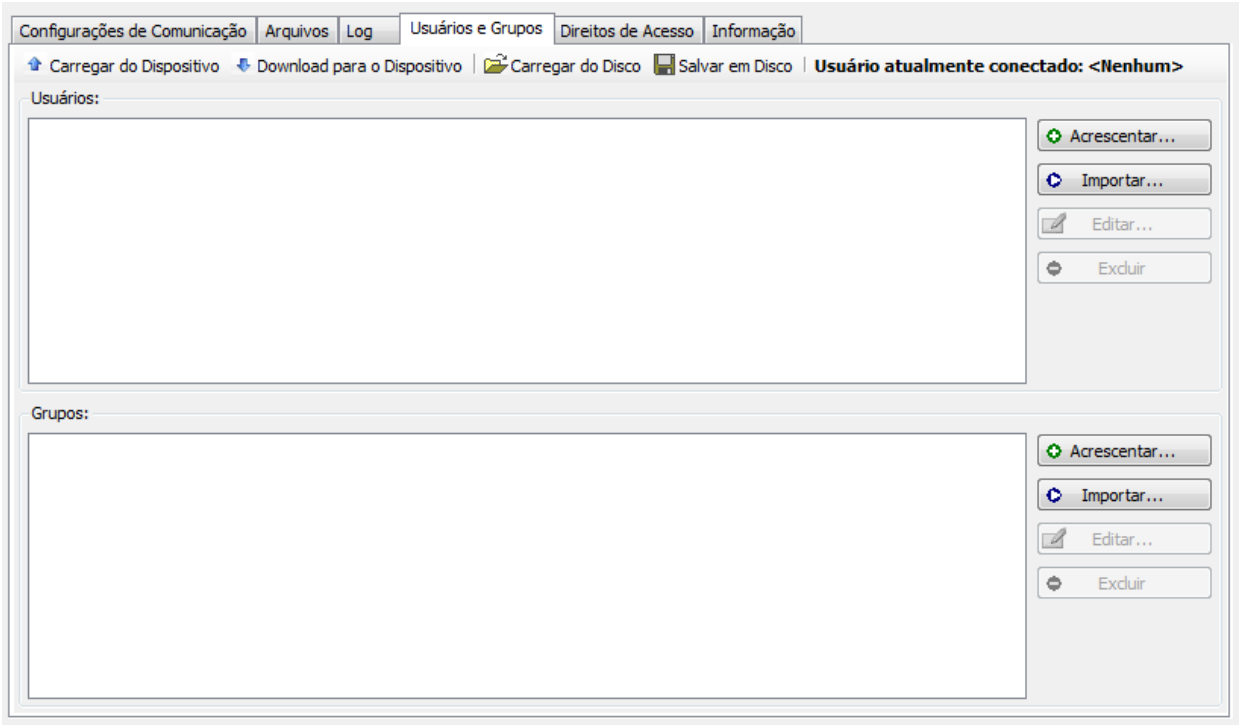


Figura 4-74. Diálogo Device, Usuários e Grupos

Usuários

Os seguintes botões estão disponíveis para configurar contas de usuários:

Acrescentar...: o diálogo *Acrescentar Usuário* define um nome de usuário e uma senha. A senha deve ser repetida no campo *Confirmar senha*.

ATENÇÃO: Ao abrir esse diálogo os campos *Senha* e *Confirmar Senha* estarão preenchidos com caracteres fictícios, o usuário deve substituir esses caracteres por uma senha válida.

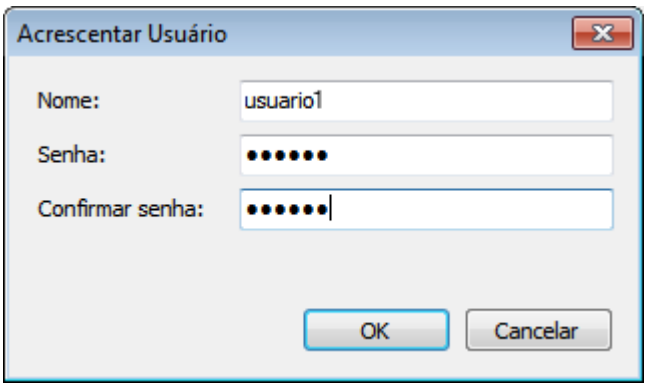


Figura 4-75. Acrescentar Usuário

Importar...: o diálogo *Importar Usuários* mostra todos os nomes de usuários atualmente definidos no gerenciamento de usuários do projeto. Selecione um ou mais itens e confirme com OK. No diálogo *Inserir Senha*, digite a senha (conforme definido no gerenciamento) para que a conta do usuário seja importada para o gerenciamento de usuário específico do dispositivo.

Editar...: o nome e senha da conta do usuário atualmente selecionada pode ser modificada. O diálogo *Editar Usuário* <nome do usuário> corresponde ao diálogo *Acrescentar Usuário*.

Excluir: a conta do usuário atualmente selecionada será excluída.

Grupos

Acrescentar...: o diálogo *Acrescentar Grupo* define um novo nome para este e seleciona, a partir dos usuários atualmente definidos, aqueles que devem fazer parte do grupo.

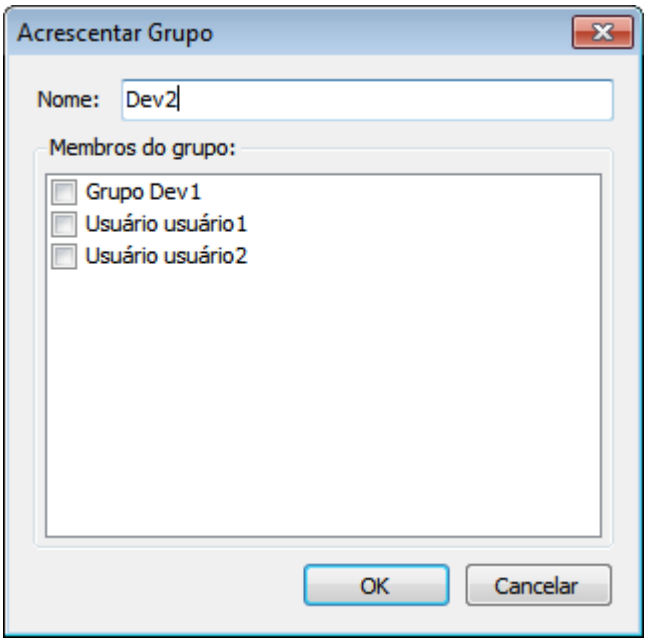


Figura 4-76. Acrescentar Grupo

Importar...: o diálogo *Importar Grupos* apresenta uma lista com os grupos atualmente definidos no gerenciamento de usuários do projeto. Selecione um ou mais itens e confirme com OK para integrá-los à lista de grupos do gerenciamento de usuários específico do dispositivo.

Editar...: o grupo atualmente selecionado pode ser modificado no que se refere ao seu nome e usuários associados. Para tanto, usa-se o diálogo *Editar Grupo* <nome do grupo>, o qual corresponde ao diálogo *Acrescentar Grupo*.

Excluir: o grupo atualmente selecionado será excluído.

Aplicando e Armazenando a Configuração Atual

Veja os botões respectivos na barra superior do diálogo.

Carregar do Dispositivo, Download para o Dispositivo: a configuração atual do gerenciamento de usuários deve ser enviada ao dispositivo para torná-la efetiva. A configuração atualmente aplicada no dispositivo pode ser carregada no diálogo de configuração.

Salvar em Disco, Carregar do Disco: a configuração atual pode ser armazenada em um arquivo xml (*.dum) e recarregada a partir deste arquivo, o que é útil para configurar a mesma configuração de usuários em vários sistemas. O diálogo padrão para procurar arquivos no sistema será fornecido com esta finalidade. O filtro do arquivo é configurado automaticamente para "*.dum" (arquivos específicos de "gerenciamento de usuários").

As configurações atuais podem ser impressas ou documentadas através dos comandos *Imprimir...* (menu *Arquivo*) e *Documentar...* (menu *Projeto*), respectivamente.

Considerações sobre Usuário e Grupos Padrão

Nas versões atuais do firmware existem os usuários: *Administrator* e *Everyone*; e os grupos: *Administrator*, *Developer*, *Everyone*, *Service* e *Watch* (Tabela 4-129).

Usuários	Grupos
Administrator	Administrator
Everyone	Developer
	Everyone
	Service
	Watch

Tabela 4-129.Usuários e Grupos

Usuários e Grupos Padrão

Os seguintes grupos e usuários estão definidos por padrão nas UCPs da Série Xtorm. Este divisão em um maior número de grupos visa apresentar uma proposta inicial de diferentes níveis de usuários que possam acessar a UCP.

Grupo Administrator

Este grupo possui todos os privilégios e não é possível removê-lo. O grupo Developer faz parte deste grupo.

Grupo Developer

Grupo criado para definir direitos de acesso a usuários que sejam desenvolvedores de aplicação. O grupo Service faz parte deste grupo. Se não for utilizado este grupo pode ser excluído.

Grupo Everyone

Este é o grupo padrão para realizar os acessos em uma UCP enquanto não existam grupos e usuários definidos.

Grupo Service

Grupo criado para definir direitos de acesso a usuários que prestem algum tipo de serviço nos CPs, como por exemplo, equipes de manutenção. O grupo Watch faz parte deste grupo. Se não for utilizado este grupo pode ser excluído.

Grupo Watch

Grupo criado para definir difeitos de acesso a usuários que apenas podem visualizar sem realizar nenhum tipo de modificação na aplicação, se não for utilizado este grupo pode ser excluído.

Usuário Administrator

O usuário Administrator está definido nos grupos Everyone e Administrator. A senha padrão do usuário Administrator é "Administrator" e pode ser modificada.

Usuário Everyone

O usuário Everyone está definido nos grupos Everyone e Administrator. Este usuário não possui senha definida.

Usuário e Grupos de Projetos Antigos

Para manter os usuários e grupos de projetos antigos em novos projetos após a atualização de firmware da UCP ou em novas UCPs Xtorm é necessário no projeto antigo com o firmware original executar o comando *Carregar do Dispositivo*, buscando assim as configurações da UCP, e após, o comando *Salvar em Disco*, salvando assim a configuração atual em um arquivo.

Na nova UCP Xtorm ou na UCP atualizada, executar o comando *Carregar do Disco*, e selecionar o arquivo gerado anteriormente, por fim, executar o comando *Download para o Dispositivo,* enviando assim as configurações para a UCP.

Direitos de Acesso

Este diálogo é fornecido em uma guia do diálogo *Device* (Editor do dispositivo). Ele faz parte do *Gerenciamento de Usuários Online* e serve para conceder e negar certas permissões ao grupo de usuários definido no momento, determinando assim os direitos de acesso a arquivos e objetos (a uma aplicação, por exemplo) quando o UCP está em execução.

Observe que estas permissões somente podem ser atribuídas a grupos e não a usuários únicos. Por isto, um usuário deve estar definido como membro de um grupo. A configuração dos usuários e grupos é feita na guia *Usuários e Grupos* do editor do dispositivo.

A Figura 4-77 mostra a permissão para adicionar e remover nós secundários para/do objeto PLC Logic concedido ao grupo de usuários *Everyone* e *Owner*.

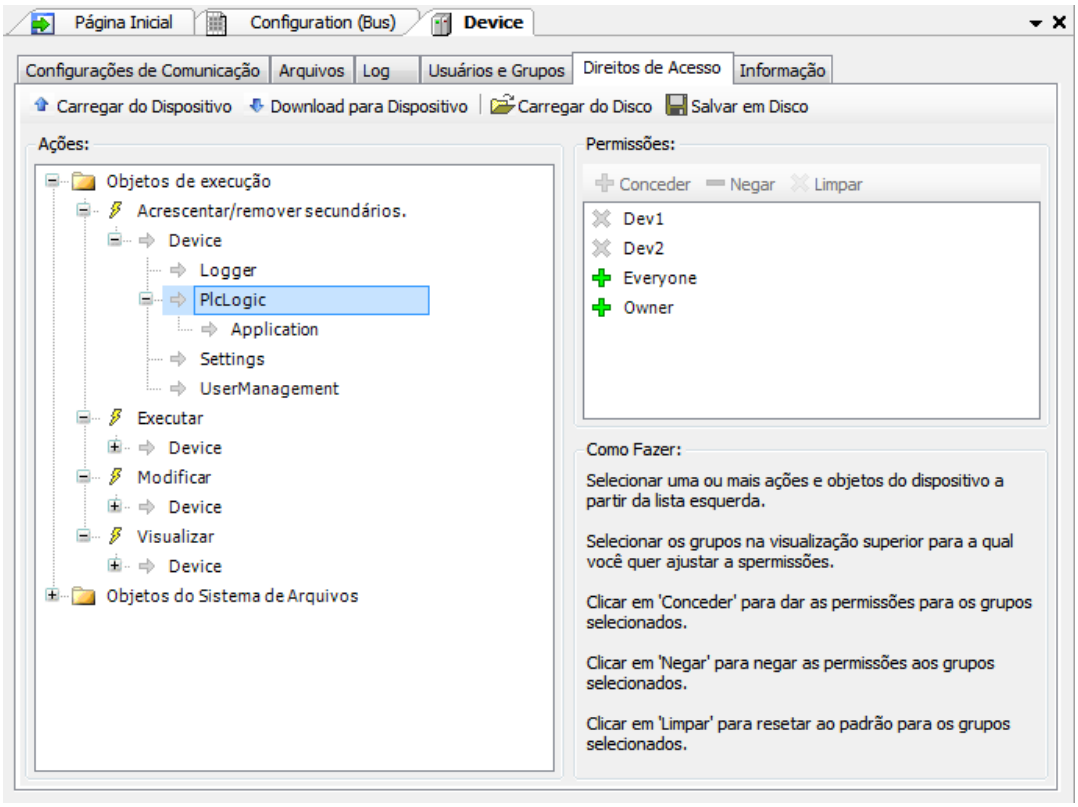


Figura 4-77. Dispositivo, Direitos de Acesso

Os direitos de acesso (acrescentar / remover secundários, executar, modificar e visualizar) são configurados para cada dispositivo e habilitam para cada usuário as ações conforme a Tabela 4-130.

Dispositivos			Ação	Direitos relevantes			
				Acrescentar / remover secundários	Executar	Modificar	Visualizar
Device	Logger		Login no device				x
			Ler entradas de Log				x
	PlcLogic	Application	Login na aplicação				x
			Download de uma aplicação em uma UCP sem aplicação	x			
			Download de uma aplicação / Alterações online			x	
			Criar Projeto de Inicialização			x	
			Visualizar variáveis				x
			Escrever valores em variáveis			x	x
			Forçar valores em variáveis			x	x
			Adicionar/Remover Breakpoint			x	
			Passar Dentro/Sobre/Fora		x		
			Executar até o Cursor		x		
			Configurar Próxima Declaração		x		
			Ler a Pilha de				x

			Chamadas				
			Iniciar/Parar aplicação		x		
			Efetuar Reset Quente/Frio/Origem			x	
	UserManagement		Visualizar configurações de usuário				x
			Modificar configurações de usuário			x	

Tabela 4-130. Ações e Direitos

Veja a seguir como definir as permissões de acesso e como fazer com que elas sejam carregadas para o dispositivo ou armazenadas em um arquivo recarregável.

Definindo as Permissões de Acesso

Para definir a permissão para executar uma ação em um ou vários objetos, selecione-os abaixo do tipo de ação desejada na janela *Ações*. Então, selecione o grupo desejado na janela *Permissões* e clique no botão *Permitir* ou *Negar* (também na janela *Permissões*).

Veja a seguir a descrição das janelas de diálogo específicas.

Ações

Esta parte do diálogo lista as ações que podem ser realizadas durante a execução em arquivos no sistema de arquivos do CP e objetos de execução, como por exemplo, aplicações. A árvore é estruturada da seguinte forma:

Categorias de objetos

No nível superior, para fins de estruturação, encontram-se as "pastas" referentes aos objetos do sistema de arquivos e objetos de execução.

Tipos de ação

Nesta "pasta", existem nós para os quatro tipos de ações passíveis de execução nos objetos específicos. Estes nós servem apenas para fins estruturais:

Adicionar/remover secundários (adição ou remoção de objetos "secundários" para um objeto existente)

Executar (por exemplo, iniciar/parar aplicações, configuração de breakpoints, etc.)

Modificar (por exemplo, envio de aplicações, etc.)

Visualizar (monitoração)

Objetos (ação "dispositivos")

Em cada nó de tipo de ação, encontram-se os "dispositivos" (objetos) da ação (por exemplo, Dispositivo).

Estes objetos mapeados na árvore de dispositivos ou na estrutura do sistema de arquivos, são exibidos de forma estruturada.

ATENÇÃO: Atribuir um direito de acesso a um "nó principal", na árvore de objetos, geralmente significa que o "nó secundário" vai herdar esta definição enquanto não receber uma definição específica própria. Entretanto, dependendo do dispositivo, isto pode ser tratado diferentemente. De qualquer forma, as heranças não são visualizadas nos diálogos.

Permissões

Este campo mostra os grupos de usuários definidos. Cada grupo é precedido por ícones que indicam a permissão atribuída no momento referente ao dispositivo atualmente selecionado na janela *Ações*.

: as ações selecionadas no momento na janela *Ações* são concedidas para o grupo.

: as ações selecionadas no momento na janela *Ações* são negadas para o grupo.

: não há definição explícita de direito de acesso para as ações selecionadas no momento, na janela *Ações*.

Se várias ações sem configurações exclusivas (em relação ao grupo) estiverem selecionadas, nenhum ícone será exibido.

Barra de botões: após ter selecionado os objetos e o grupo desejados (janela *Ações e Permissões*), um dos seguintes botões pode ser usado:

Conceder : permissão de acesso explicitamente concedida.

Negar : permissão de acesso explicitamente negada.

Limpar : o direito de acesso concedido para as ações atualmente selecionadas na janela *Ações* será apagado, ou seja, ele retornará para o padrão.

Aplicando e Armazenando a Configuração Atual

Veja os botões respectivos na barra superior do diálogo.

Carregar do Dispositivo , Download para o Dispositivo : as definições dos direitos de acesso configuradas devem ser enviadas para o dispositivo para se tornarem efetivas. As configurações atualmente aplicadas no dispositivo são carregadas no diálogo de configuração.

Salvar em Disco , Carregar do Disco : a configuração atual pode ser armazenada em um arquivo-xml (*.drm) e recarregada a partir deste arquivo, o qual é útil para definir a mesma configuração de usuário em vários sistemas. Para tanto, será fornecido o diálogo padrão para busca no sistema de arquivo. O filtro do arquivo automaticamente é configurado para "*.drm" (arquivos de "direitos de acesso de dispositivo").

As configurações atuais podem também ser documentadas em versões impressas via comando *Imprimir* (menu *Arquivo*) ou *Documentar* (menu *Projeto*).

Direitos de Acesso de Projetos Antigos

Para manter os direitos de acesso de projetos antigos em novos projetos após a atualização de firmware da UCP ou em novas UCPs Xtorm é necessário no projeto antigo com o firmware original executar o comando *Carregar do Dispositivo*, buscando assim as configurações da UCP, e após, o comando *Salvar em Disco*, salvando assim a configuração atual em um arquivo.

Na nova UCP Xtorm ou na UCP atualizada, executar o comando *Carregar do Disco*, e selecionar o arquivo gerado anteriormente, por fim, executar o comando *Download para o Dispositivo*, enviando assim as configurações para a UCP.

5. Programação Inicial

O objetivo deste capítulo é auxiliar na programação e configuração da UCP da Série Hadron Xtorm, para que o usuário de os primeiros passos antes de iniciar a programação de uma UCP.

As UCPs da Série Hadron Xtorm usam as linguagens padrão IEC 61131-3, IL, ST, LD, SFC e FBD. IL e ST são linguagens textuais similares a Assembly e linguagem C respectivamente. LD, SFC e FBD são linguagens gráficas. O LD usa a representação de relés e blocos e é similar a diagramas de relé. O SFC usa a representação de um diagrama sequencial e permite uma visualização clara das funções executadas em cada ação. As UCPs desta Série oferecem também uma sexta linguagem chamada CFC.

A programação é feita na interface MasterTool Xtorm. O MasterTool Xtorm possibilita o uso de seis linguagens no mesmo projeto, fornecendo, assim, os melhores recursos que cada linguagem pode oferecer. Como resultado disto, o usuário tem eficientes desenvolvimentos de aplicações e facilidade na documentação e em futuras manutenções.

ATENÇÃO: A configuração física bem como a ordem de disposição dos módulos no(s) barramento(s) deve ser a mesma configurada via MasterTool para que não sejam gerados diagnósticos e o sistema tenha condições de funcionamento correto.

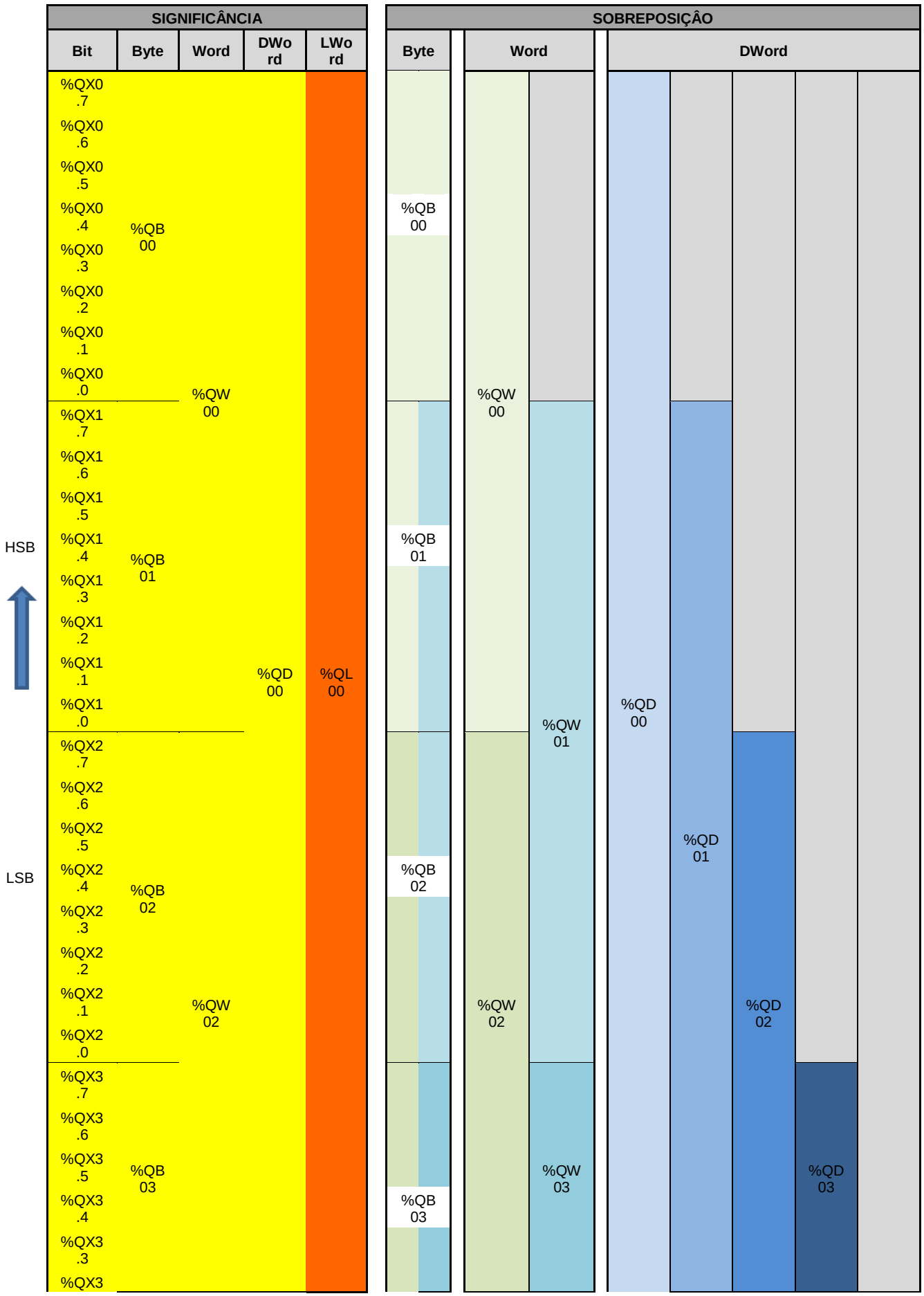
Organização e Acesso à Memória

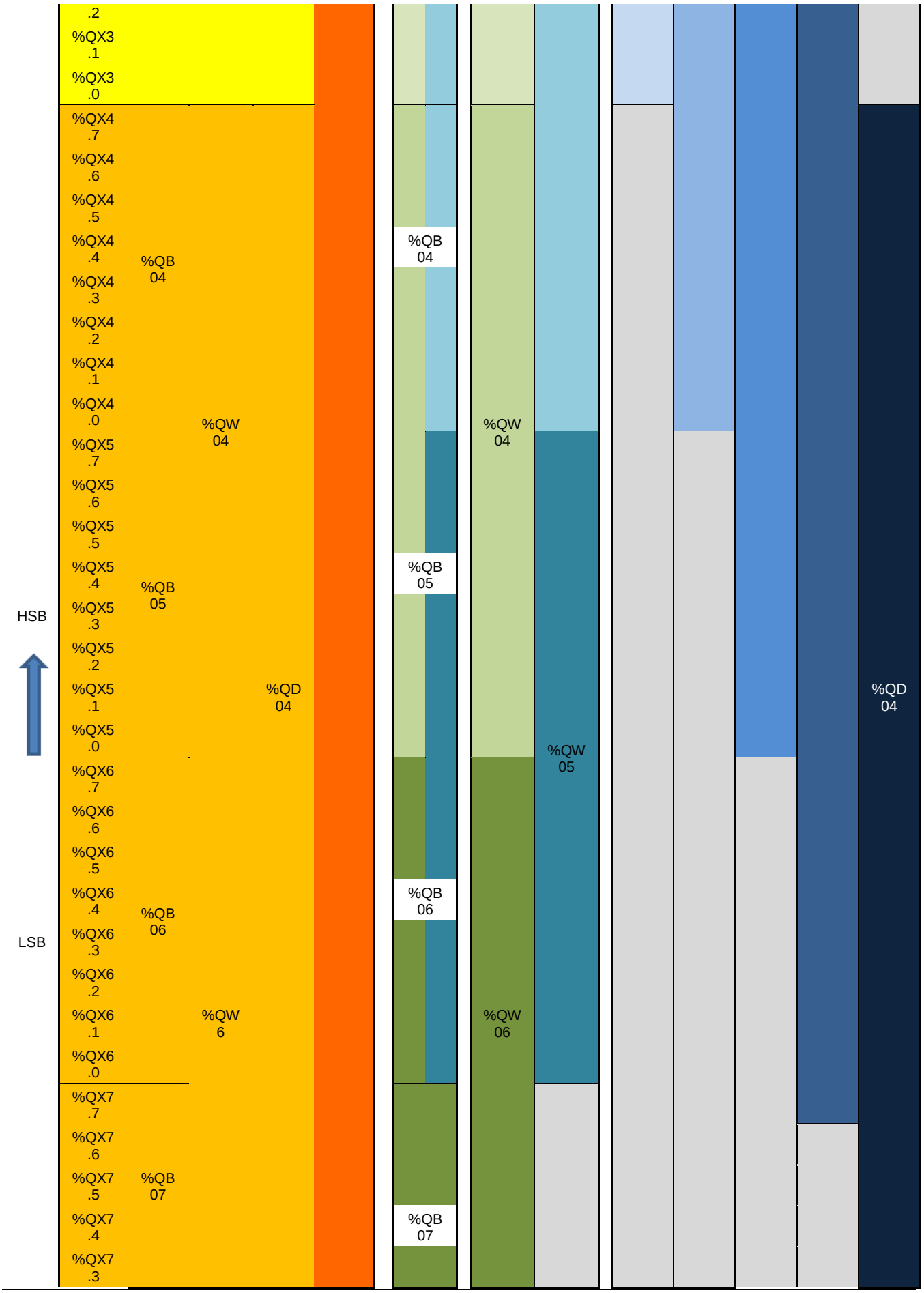
A Série Hadron Xtorm utiliza uma inovadora característica de organização e acesso à memória denominada big-endian, ou seja, o octeto mais significativo é armazenado primeiramente e sempre será o de menor endereço (Ex.: %QB0 sempre será mais significativo que o %QB1, como na Tabela 5-1, que utiliza a palavra UCPHADRON XTORM, onde a letra U é o octeto 0 e a letra O é o octeto 7).

O acesso à memória deve ser realizado com cuidado pois, variáveis com maior número de bits (WORD, DWORD, LONG), utilizam como índice o octeto mais significativo, ou seja, a %QD4 terá, como octeto mais significativo, o octeto %QB4. Não sendo necessário realizar cálculos para saber qual é a DWORD correspondente a determinados bytes. Organização little e big endian está na Tabela 5-1.

HSB <– Little-endian (Traditional) –> LSB								
BYTE	**%QB7**	**%QB6**	**%QB5**	**%QB4**	**%QB3**	**%QB2**	**%QB1**	**%QB0**
	U	C	P	X	T	O	R	M
WORD	**%QW3**		**%QW2**		**%QW1**		**%QW0**	
	UC		PX		TO		RM	
DWORD	**%QD1**				**%QD0**			
	UCPX				TORM			
LWORD	**%QL0**							
	UCP XTORM							
HSB <– Big-endian (HADRON XTORM) –> LSB								
BYTE	**%QB0**	**%QB1**	**%QB2**	**%QB3**	**%QB4**	**%QB5**	**%QB6**	**%QB7**
	U	C	P	X	T	O	R	M
WORD	**%QW0**		**%QW2**		**%QW4**		**%QW6**	
	UC		PX		TO		RM	
DWORD	**%QD0**				**%QD4**			
	UCPX				TORM			
LWORD	**%QL0**							
	UCPXTORM							

Tabela 5-1. Exemplo


SIGNIFICÂNCIA
Bit
Byte
Word
DWord
LWord
%QX0.7
%QX0.6
%QX0.5
%QX0.4
%QX0.3
%QX0.2
%QX0.1
%QX0.0
%QB00
%QW00
%QX1.7
%QX1.6
%QX1.5
%QX1.4
%QX1.3
%QX1.2
%QX1.1
%QX1.0
%QB01
%QD00
%QL00
%QX2.7
%QX2.6
%QX2.5
%QX2.4
%QX2.3
%QX2.2
%QX2.1
%QX2.0
%QB02
%QW02
%QX3.7
%QX3.6
%QX3.5
%QX3.4
%QX3.3
%QX3
%QB03
HSB
LSB
SOBREPOSIÇÃO
Byte
Word
DWord
%QB00
%QW00
%QB01
%QW01
%QD00
%QB02
%QW02
%QD01
%QD02
%QB03
%QW03
%QD03


.2
%QX3
.1
%QX3
.0
%QX4
.7
%QX4
.6
%QX4
.5
%QX4
.4
%QX4
.3
%QX4
.2
%QX4
.1
%QX4
.0
%QX5
.7
%QX5
.6
%QX5
.5
%QX5
.4
%QX5
.3
%QX5
.2
%QX5
.1
%QX5
.0
%QX6
.7
%QX6
.6
%QX6
.5
%QX6
.4
%QX6
.3
%QX6
.2
%QX6
.1
%QX6
.0
%QX7
.7
%QX7
.6
%QX7
.5
%QX7
.4
%QX7
.3
%QB
04
%QW
04
%QB
05
%QD
04
%QB
06
%QW
6
%QB
07
HSB
LSB
%QB
04
%QW
04
%QB
05
%QW
05
%QB
06
%QW
06
%QB
07
%QD
04

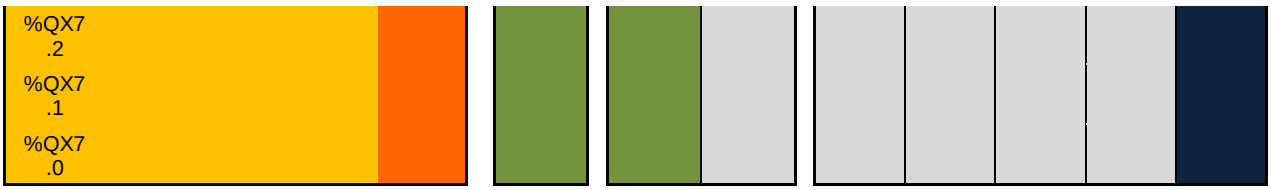


Tabela 5-2. Organização e Acesso à Memória

A Tabela 5-2 mostra a organização e acesso à memória, exemplificando a significância dos bytes (Significance) e a disposição dos demais tipos de variável, inclusive a sobreposição (Overlapping):

Perfis de Projeto

Um perfil de projeto no MasterTool Xtorm é um conjunto de regras, características comuns e padrões utilizados no desenvolvimento de uma solução de automação , um perfil influencia a forma de implementação da aplicação. Seguir um perfil é uma forma de reduzir a complexidade na programação. As aplicações podem ser criadas conforme um dos seguintes perfis:

- UTR
- Personalizado

O MasterTool Xtorm pode disponibilizar templates compatíveis para cada perfil definido para o RTS. Quando o usuário seleciona um template como modelo na criação de um projeto, a nova aplicação será desenvolvida conforme um determinado perfil, adotando as regras, características e padrões definidos pelo perfil associado ao template. Cada perfil de projeto define nomes padronizados para as tarefas e programas, os quais são pré-criados pelos templates de projeto. O desenvolvedor é obrigado a seguir rigorosamente a nomenclatura para tarefas, mas pode seguir ou não os nomes sugeridos para os respectivos programas.

Para garantir a compatibilidade de um projeto a um determinado perfil ao longo do desenvolvimento, são utilizadas duas abordagens:

- O MasterTool Xtorm somente permite a criação de projetos baseados em um template, selecionando ao mesmo tempo o perfil a ser utilizado
- Na geração de código, o MasterTool Xtorm realiza a verificação de todas as regras definidas para o perfil válido para o projeto

As próximas seções detalham as características ou padrões de cada perfil de projeto. Com base nestas definições, recomenda-se que o usuário sempre procure utilizar o perfil mais simples que atenda as necessidades da sua aplicação, migrando para o outro mais sofisticado apenas quando as regras correspondentes estiverem sendo mais entraves ao desenvolvimento do que simplificações didáticas. Cabe ressaltar que a ferramenta de programação permite a alteração do perfil de um projeto existente, mas caberá ao desenvolvedor realizar qualquer ajuste necessário para que o projeto se torne compatível com as regras do novo perfil selecionado.

ATENÇÃO: no decorrer dos perfis de projeto são nomeados alguns tipos de tarefas, as quais estão descritas na seção Configurando as Tarefas.

Perfil para UTR

O “Perfil para UTR” pode ser tanto criado para projetos redundantes, quanto para projetos não redundantes. Nesse tipo de perfil, a aplicação pode ter até duas tarefas de usuário, a “MainTask” que é sempre criada como padrão para projetos simples e redundantes, e a “ProtTask” que somente será criada se o usuário definir seu uso no projeto. A tarefa “MainTask” é responsável pela execução de uma única unidade de programação do tipo “Program” denominada “MainPrg”. Este programa não pode ser editado e é responsável por chamar outras quatro unidades de programação do tipo “Program”, denominadas “StartPrg”, “EngineeringPrg”, “AlarmPrg” e “UserPrg”.

Os programas “StartPrg” e “UserPrg” são os únicos da tarefa “MainTask” que podem conter código de usuário podendo chamar outras unidades de programação do tipo “Program”, “Function” ou “Function Block”, mas o código de usuário será executado pela tarefa “MainTask”. Já os programas “EngineeringPrg” e “AlarmPrg” são criados automaticamente e são responsáveis, respectivamente, por realizar as conversões de engenharia e sinalizar os alarmes mapeados pelo usuário nas telas de configuração da UCP. Para mais informações sobre alarmes e conversão de engenharia, consulte o capítulo Configuração da UCP.

A “ProtTask” também é responsável por executar uma única unidade de programação do tipo “Program”, cujo nome é “ProtPrg”. Este programa não pode ser editado e é responsável por chamar a “UserProtPrg” que, por sua vez, pode chamar outras unidades de programação e é responsável por executar as rotinas de proteção do projeto.

Neste perfil, a tarefa “MaintTask” será do tipo cíclica “Cyclic” com prioridade fixada como 13 (treze) e executa exclusivamente o programa “MainPrg”, enquanto a tarefa “ProtTask” também será do tipo cíclica, mas terá sua prioridade fixada como 1 (um), sendo assim, mais prioritária que a anterior, e executa exclusivamente o programa “ProtPrg”.

As tarefas “MainTask” e “ProtTask” já estão completamente definidas e o desenvolvedor precisa criar o programa “StartPrg”, “UserPrg” e “UserProtPrg” optando por qualquer uma das linguagens da norma IEC 61131-3. Nem sempre é possível converter um programa para outra linguagem, mas sempre é possível criar um novo programa com o mesmo nome em substituição que seja construída em linguagem diversa.

A opção padrão do Mastertool Xtorm é utilizar o “Mastertool Standard Project” associado ao perfil “Perfil para UTR”, o qual inclui os programas “StartPrg”, “UserPrg” e “UserProtPrg” criados nas linguagens escolhidas durante a criação do projeto, e os programas “ProtPrg”, “MainPrg”, “AlarmPrg” e “EngineeringPrg” sempre na linguagem ST (Structured Text).

Tarefa	POU	Prioridade	Tipo	Valor Padrão	Opções	Evento
MainTask*	MainPrg	13	Cíclica	100 ms	5 a 100 ms	-
ProtTask	ProtPrg	1	Cíclica	4 ms	4 a 20 ms	-

Tabela 5-3. Tarefa no Perfil para UTR

* Com redundância de UCP, o tempo de ciclo mínimo da MainTask é de 50 ms. Veja mais informações sobre redundância no capítulo Redundância com UCP HX3040.

Perfil Personalizado

O perfil de projeto “Personalizado” permite ao desenvolvedor explorar todas as potencialidades do Runtime System implantado na unidade central de processamento da Série Xtorm. Nenhuma das funcionalidades é desabilitada, nenhuma prioridade, associação entre tarefas e programas é imposta. Também é possível criar tarefas e programas com quaisquer nomes, porém, a “MainTask” e os programas “StartPrg”, “MainPrg”, “AlarmPrg” e “EngineeringPrg”, devem sempre existir com estes nomes neste perfil, além de que, os programas “MainPrg”, “AlarmPrg” e “EngineeringPrg” não podem ser editados.

Além das tarefas em tempo real com prioridade 00 a 15, as quais são escalonadas por prioridade, neste perfil também é possível definir tarefas com prioridades menores na faixa de 16 a 31.

No perfil “Personalizado”, o usuário pode gerar algumas tarefas que não estão presentes no perfil “Perfil para UTR”. Duas destas outras tarefas são denominadas “CyclicTask00”, “CyclicTask01” cada qual responsável pela execução do respectivo programa “CyclicPrg<nn>”. As tarefas “CyclicTask<nn>” são do tipo cíclica “Cyclic”. Estes dois tipos formam um conjunto denominado de tarefas básicas.

Este perfil pode adicionalmente incluir tarefas de interrupção com maior prioridade que as tarefas básicas e que consequentemente podem interromper (preemptar) a execução daquelas a qualquer momento.

Quanto ao tempo de cão-de-guarda o ideal é manter o tempo médio de execução das tarefas em, no máximo, 50% do tempo de cão-de-guarda. Assim, diminuem as chances de ocorrerem erros de cão-de-guarda por eventuais picos de tempo na execução da tarefa. Este parâmetro somente é editável no perfil personalizado.

As tarefas chamadas “ExternInterruptTask<nn>” são tarefas de interrupção do tipo Externa “Extern” cuja execução é disparada por algum evento externo, tais como variação de um sinal de controle numa porta serial ou variação de uma entrada digital no barramento. Essas tarefas são responsáveis pela execução dos programas “ExternInterruptPrg<nn>”. As tarefas chamadas “TimeinterruptTask<nn>” são tarefas de interrupção do tipo cíclica “Cyclic”, sendo responsáveis pela execução dos programas “TimeInterruptPrg<nn>”.

Neste perfil de projeto, a aplicação ainda pode incluir a tarefa de usuário “FreeTask” do tipo “Freewheeling”, responsável pela execução do programa “FreePrg”. Esta é uma tarefa de baixa prioridade e pode ser interrompida por todas as demais, a mesma é capaz de executar códigos que podem ficar bloqueados.

No modelo “Perfil Personalizado”, existem oito tarefas e suas POUs já completamente definidas conforme apresentado, bem como os respectivos programas associados criados na linguagem que o usuário selecionar. Os intervalos, eventos de disparo e prioridades de qualquer tarefa podem ser configurados pelo usuário.

.

Tarefa	POU	Prioridade	Tipo	Valor Padrão	Opções	Evento
MainTask	MainPrg	13	Cíclica	100 ms		-
CyclicTask00	CyclicPrg00	13	Cíclica	200 ms		-
CyclicTask01	CyclicPrg01	13	Cíclica	500 ms		-
ExternInterruptTask00	ExternInterruptPrg00	02	Externa	-		IO_INT_0
TimeInterruptTask00	TimeInterruptPrg00	01	Cíclica	20 ms		-
ExternInterruptTask01	ExternInterruptPrg01	11	Externa	-		IO_INT_1
TimeInterruptTask01	TimeInterruptPrg01	09	Cíclica	30 ms		-
FreeTask	FreePrg	31	Contínua	-		-

Tabela 5-4. Tarefas no Perfil Personalizado

ATENÇÃO: Os nomes sugeridos para as POUs associadas às tarefas não são consistidos. Os mesmos podem ser substituídos desde que sejam substituídos também nas configurações das tarefas.

Novo Projeto

Como descrito anteriormente, existem diferentes tipos de perfis de projeto, sendo dependentes do conhecimento técnico de cada usuário. Neste capítulo, será abordada somente a criação de um novo projeto a partir do perfil Simples, utilizando a ferramenta “Wizard”, a qual apresenta as opções de configuração do sistema ao usuário.

Inicialmente, o usuário deverá criar um novo projeto no MasterTool Xtorm a partir do menu Arquivo e logo em seguida, “New Project...”, conforme mostra a Figura 5-1:

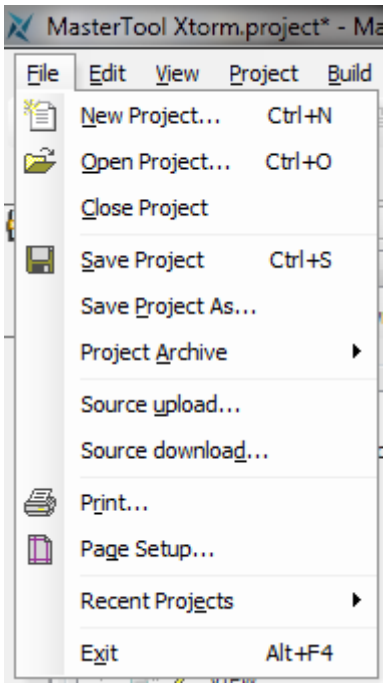


Figura 5-1. Novo Projeto

Após, uma janela será apresentada ao usuário, solicitando que o mesmo selecione o tipo de projeto que deseja fazer e, em seguida, escreva um nome e a localização para armazenar o projeto no computador. Clicar em OK para prosseguir ou Cancel para interromper.

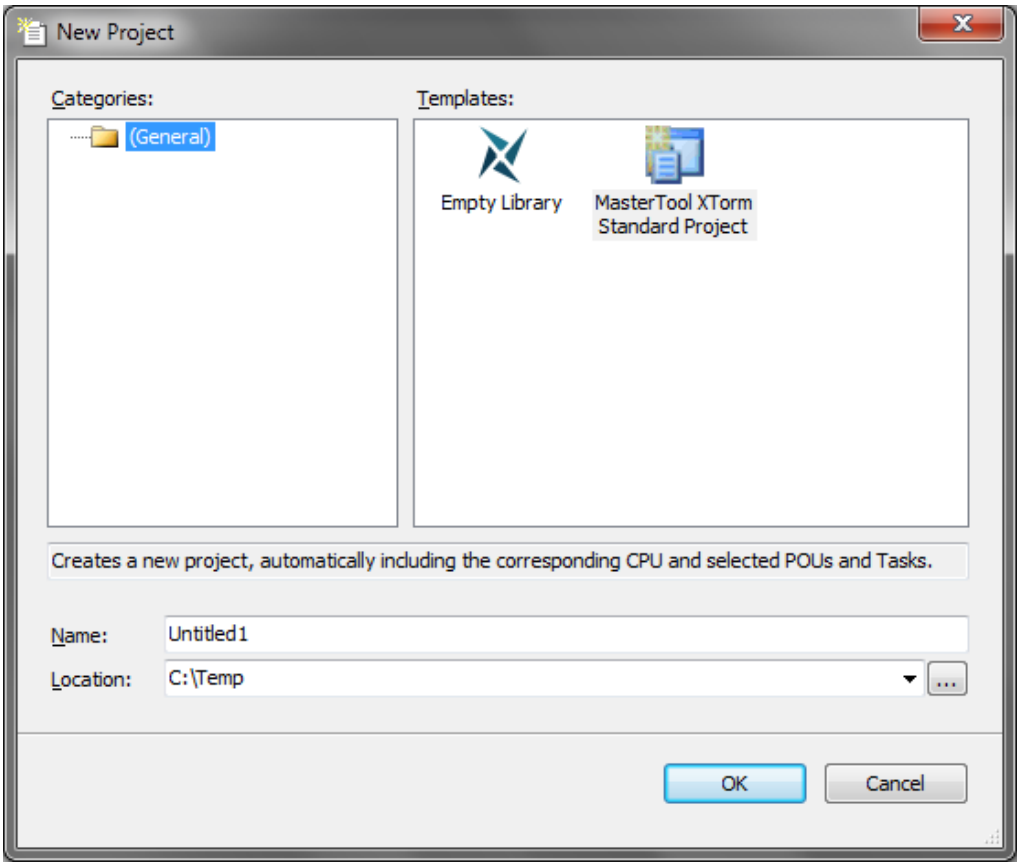


Figura 5-2. Classificação do Projeto

A seguir, o usuário deverá selecionar a UCP desejada, os módulos de hardware básicos que compõem o barramento, ou seja, o modelo de bastidor e de fonte de alimentação e a configuração de redundância. Nesse caso, será utilizada a UCP HX3040 (sem redundância), um bastidor HX9001 e uma fonte de alimentação HX8320 (sem redundância).

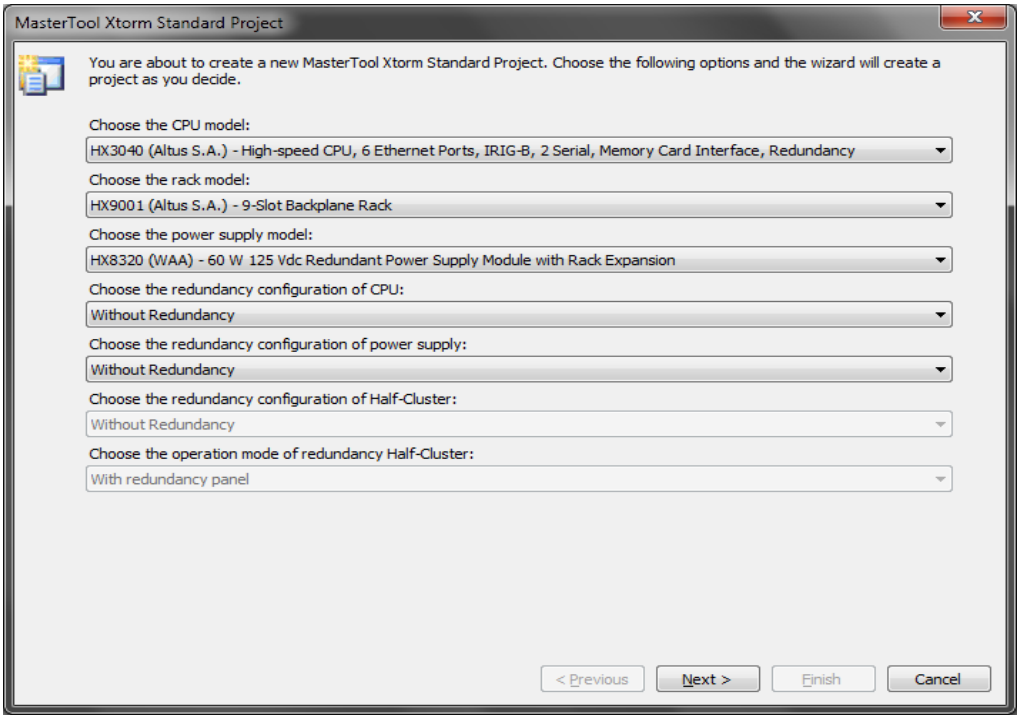


Figura 5-3. Opções do Projeto

Caso o usuário selecione a opção para redundância de UCP, a opção para redundância de fonte de alimentação será automaticamente marcada. Esta é uma característica de projeto, sendo que para projetos com redundância de UCP, será obrigatoriamente necessária à redundância de fonte de alimentação. Na Figura 5-4 podemos visualizar uma tela com a opção de redundância de UCP selecionada.

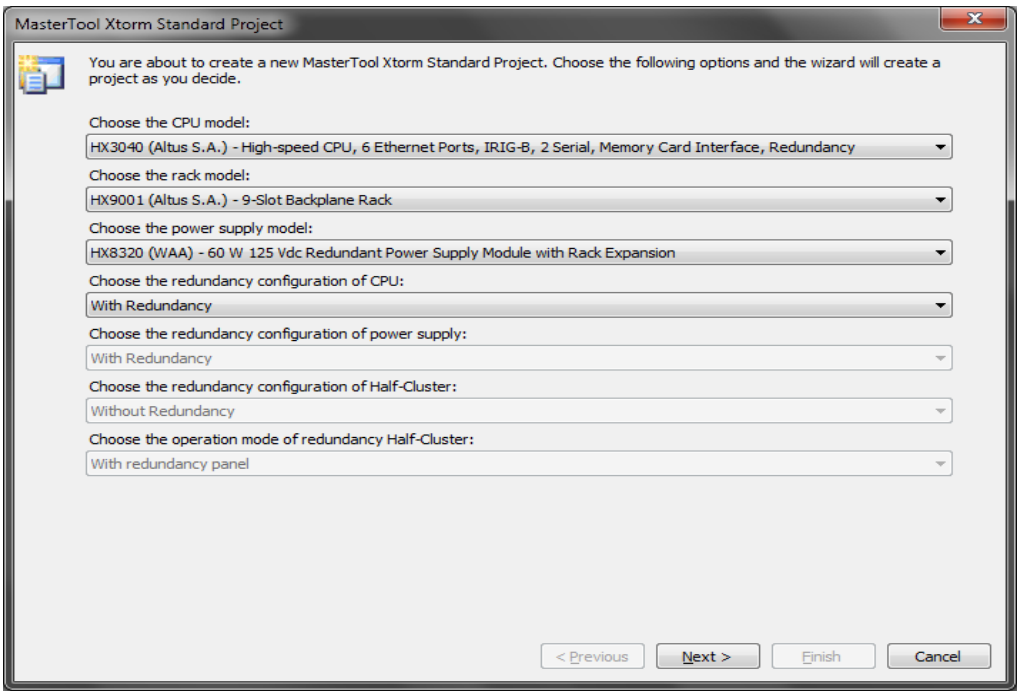


Figura 5-4. Opções de projeto com redundância de UCP habilitada

Na próxima tela, o usuário deverá configurar a quantidade de módulos que usará no projeto, sendo assim o Wizard criará os objetos destes módulos dentro do projeto automaticamente.

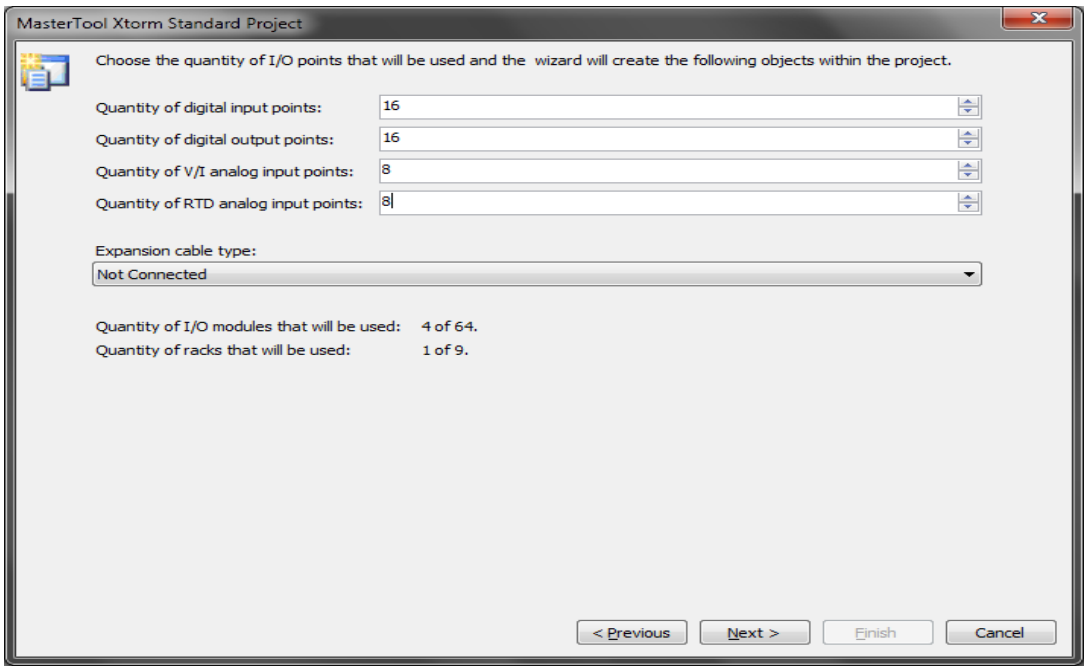


Figura 5-5. Opções de módulos de E/S do Projeto

Então, o usuário deverá selecionar o perfil para o projeto, e a linguagem padrão para as POUs (programas). Nesse caso o novo projeto está exemplificado perfil RTU sem redundância e linguagem ST. Clicar em Next para prosseguir ou Cancel para interromper a criação do projeto.

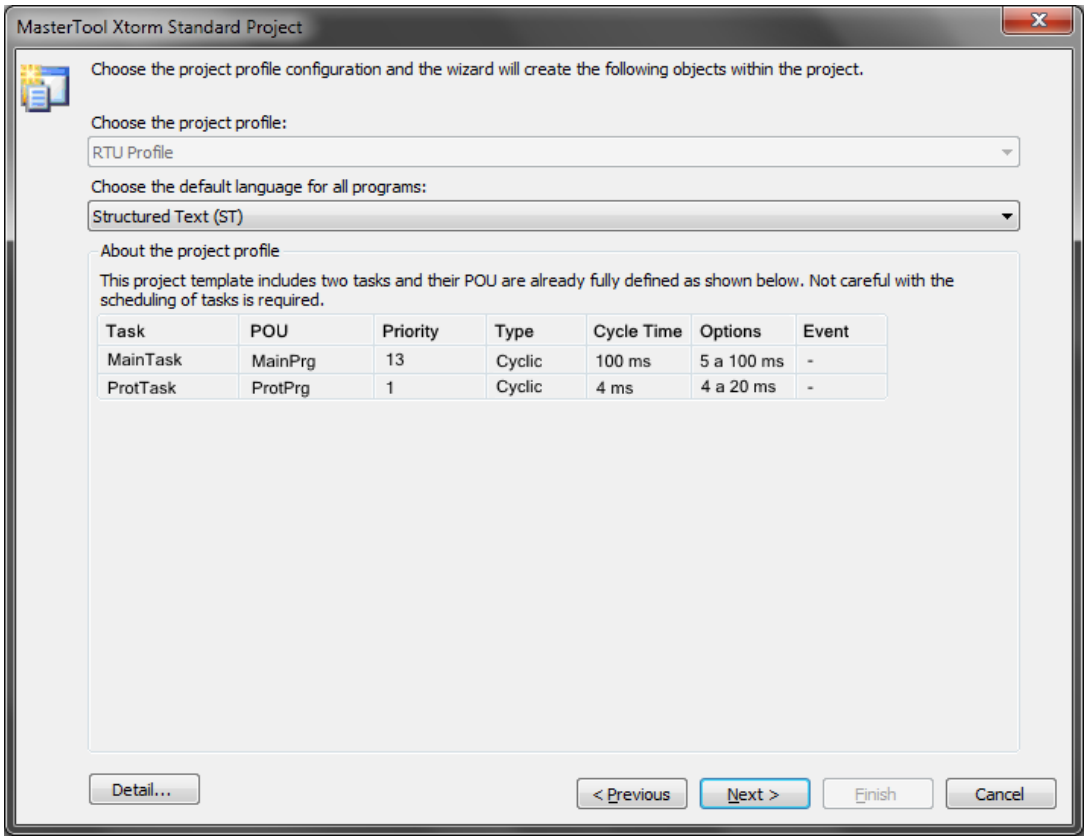


Figura 5-6. Seleção das Características de Perfil do Projeto

A próxima tela define a linguagem da POU criada pelo perfil selecionado. Como o projeto criado não é redundante, existem apenas duas POUs (UserPrg) e (StartPrg) a linguagem ST foi mantida. Clicar em Anterior para voltar à tela antecedente, Finish para finalizar ou Cancel para interromper.

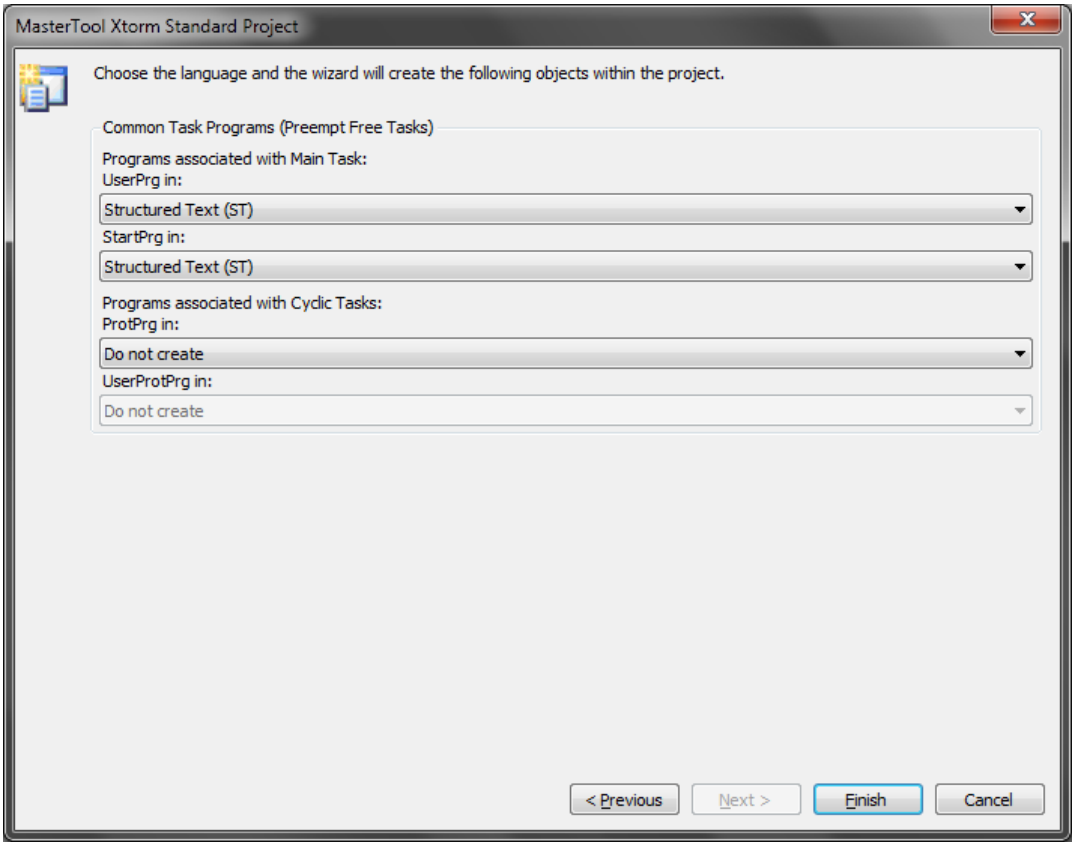


Figura 5-7. Linguagem de Programação

Ao pressionar o botão Concluir, o MasterTool Xtorm iniciará a criação do ambiente de desenvolvimento do projeto. Esse procedimento pode levar alguns segundos.

Acrescentando Módulos

Por padrão, a UCP e os módulos de hardware selecionados na criação do projeto já são inseridos na configuração do sistema. Caso necessário, o usuário ainda pode incluir outros módulos necessários. A UCP Xtorm pode ter até 64 módulos de entradas e saídas distribuídos num total de 9 bastidores. Para mais informações consulte a CT123200 da fonte HX8320 Xtorm.

Caso a aba Biblioteca de Produtos não esteja disponível na tela do MasterTool Xtorm, a mesma deve ser incluída, através do menu Visualizar, clicando sobre o item “Product Library”, conforme mostra a Figura 5-8:

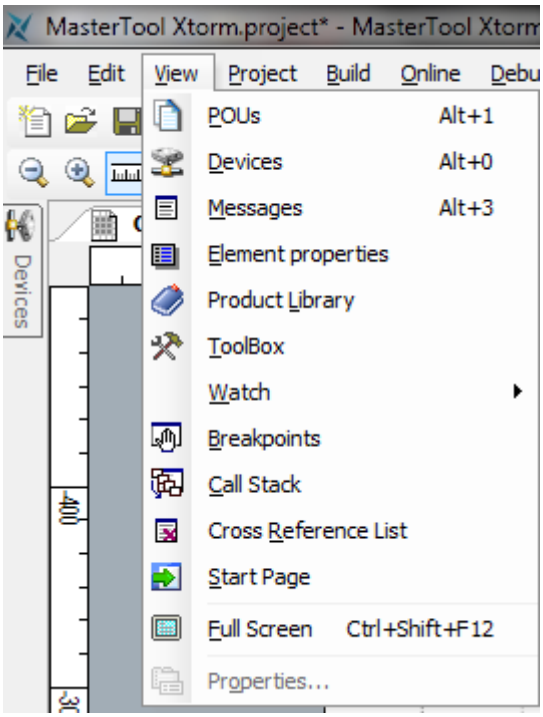


Figura 5-8. Visualizando a Biblioteca

Após, basta selecionar o módulo a ser inserido no projeto e, mantendo pressionada a tecla esquerda do mouse, arrastá-lo para a área de configuração do barramento.

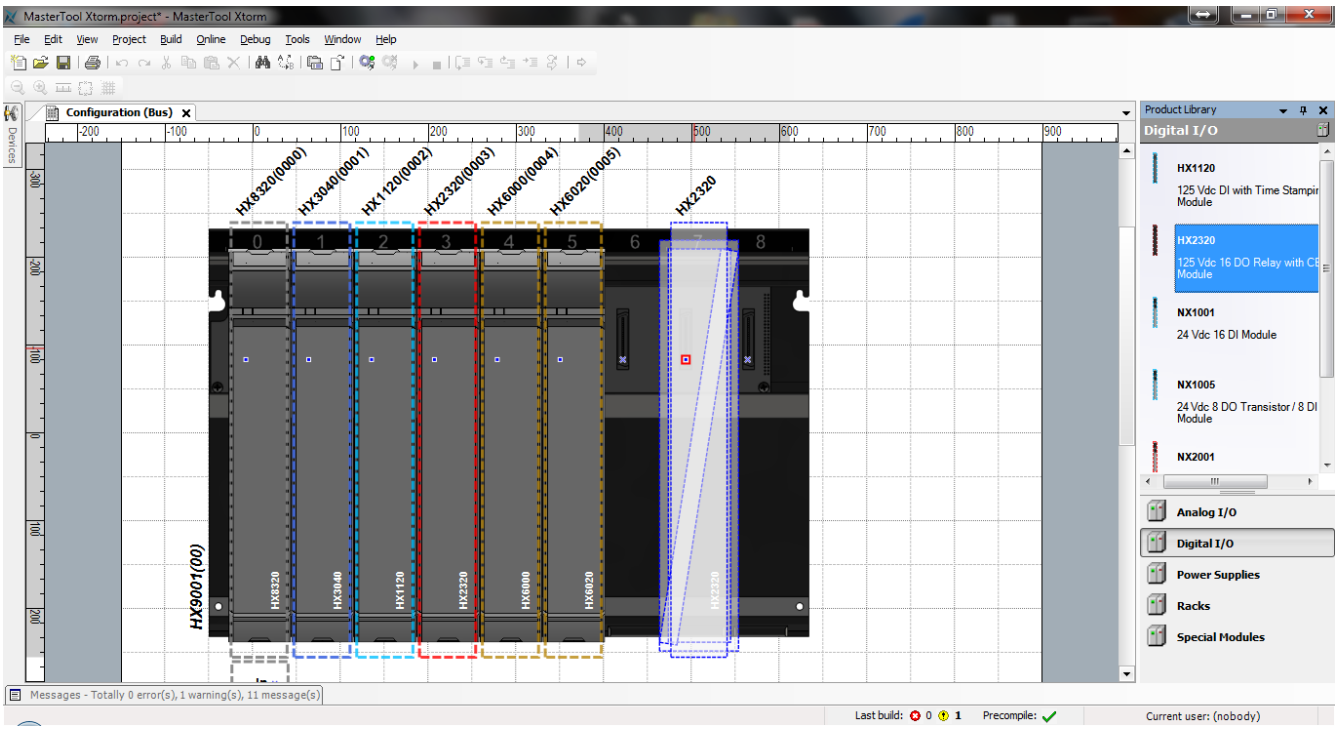


Figura 5-9. Acrescentando Módulos

Criando POUs

Uma POU (Program Organization Unit, ou Unidade de Organização de Programa), é uma subdivisão do programa aplicativo que pode ser escrito em qualquer uma das linguagens disponíveis no software MasterTool Xtorm.

Com a criação do projeto através de um perfil selecionado, algumas POUs já são criadas, porém o usuário poderá criar quantas quiser, limitado pelo tamanho máximo da memória de programa.

Para inserir uma nova POU, basta clicar com o botão direito sobre o nome da aplicação, (nome padrão criado para a aplicação é “Application”) , selecionar “Add Object” e, então, “POU...”, conforme mostra a Figura 5-10:

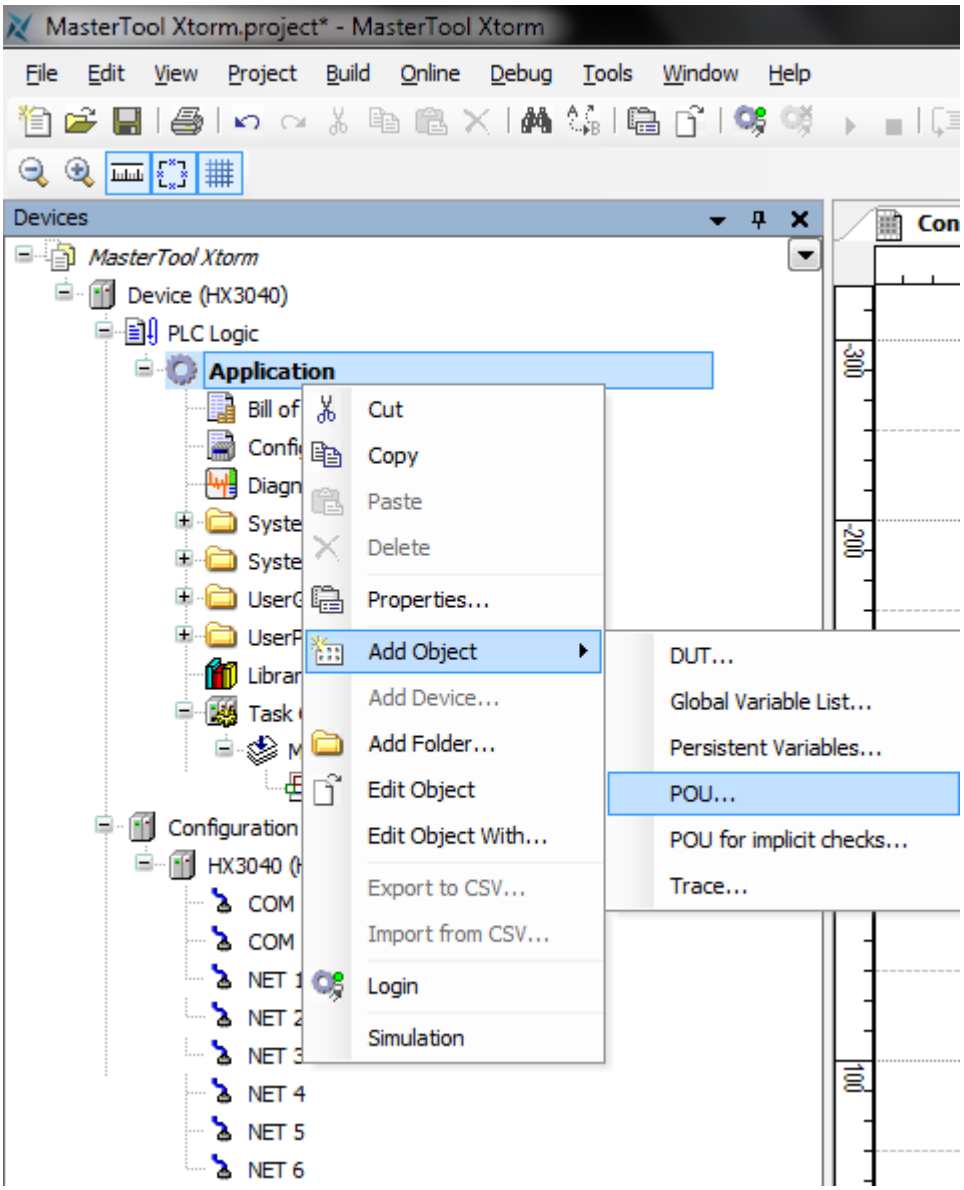


Figura 5-10. Inserindo POUs

Uma janela de configuração irá surgir na tela, na qual o usuário deve colocar o nome da POU, selecionar o tipo e a linguagem que se deseja implementar. A seguir, deve clicar em Add.

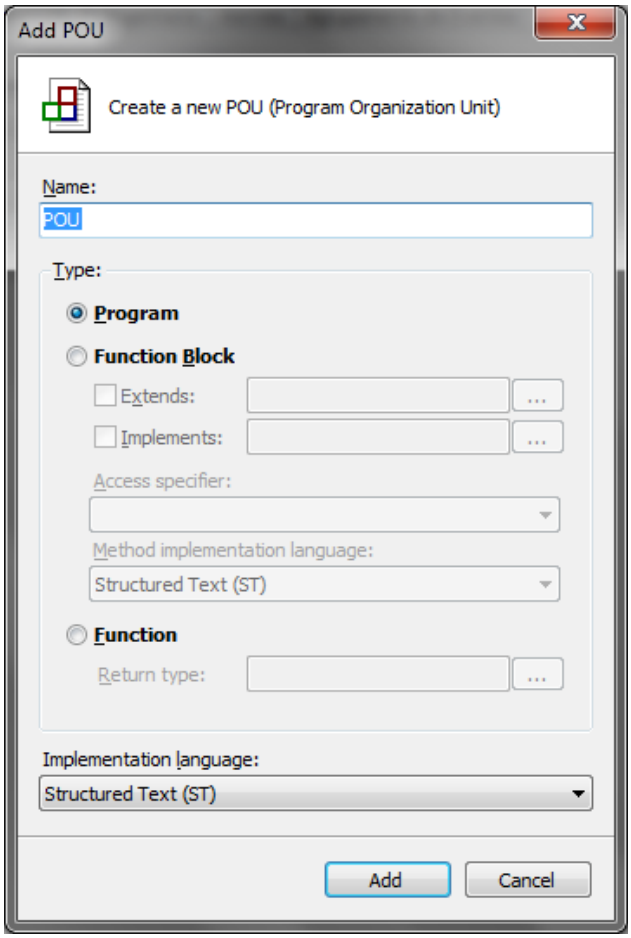


Figura 5-11. Classificando a POU

Para editar a POU basta selecionar a aba com o nome correspondente e iniciar o desenvolvimento da aplicação na linguagem escolhida anteriormente. O mesmo procedimento é válido para as POUs criadas automaticamente pelo perfil do projeto.

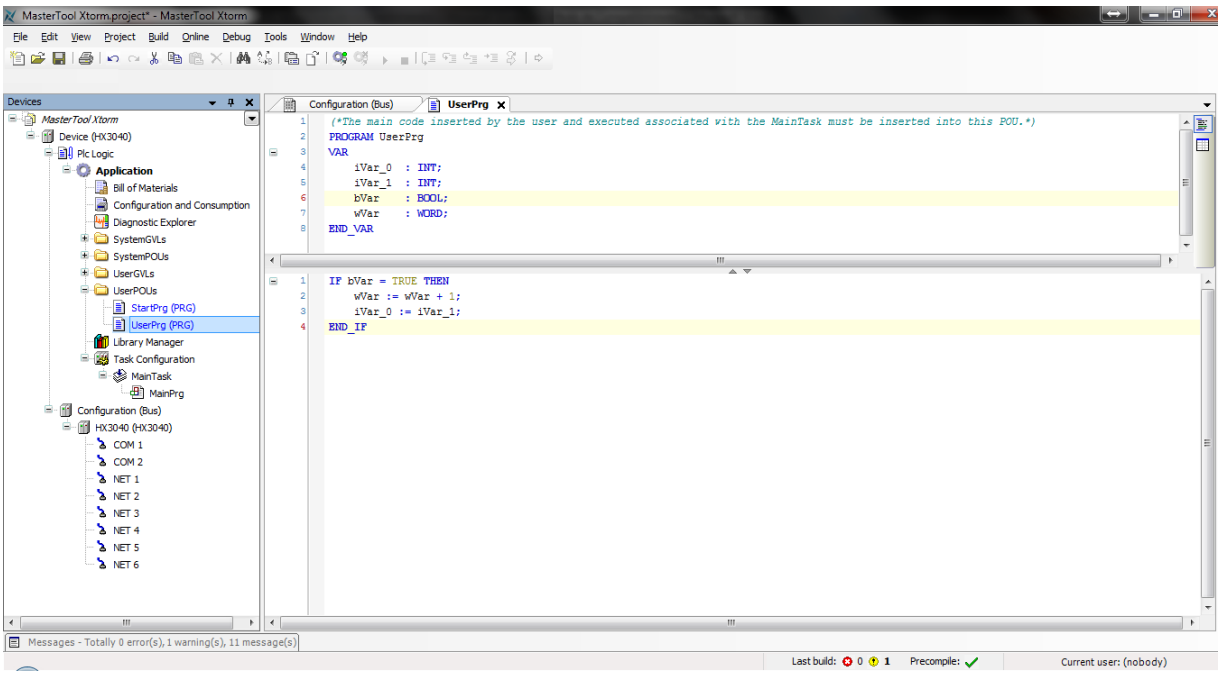


Figura 5-12. Editando a POU

Criando Tarefas

Uma POU deve estar vinculada a alguma tarefa, para que possa ser executada. Esse mecanismo de escalonamento, denominado de Tarefa, é muito útil para sistemas de tempo real que definem a execução periódica ou em resposta a um evento (mudança de estado de alguma variável booleana). As tarefas controlam a execução de programas em diferentes taxas, em função das características da aplicação. A necessidade de executar programas em taxas diferentes tem por objetivo às exigências de tempo de resposta do processo sob controle e otimizar o uso da capacidade de processamento da UCP. Os controladores que fazem o uso de tarefas são denominados sistemas multitarefa.

Somente será permitida a criação de novas tarefas quando o perfil de projeto selecionado for o Personalizado, sendo que nos demais perfis as tarefas possíveis são criadas e configuradas automaticamente.

Desta forma, para incluir uma nova tarefa (caso o perfil selecionado permita), basta clicar com o botão direito sobre Configuração de tarefas, selecionar “Add Object” e então “Task...”, conforme mostra a Figura 5-13:

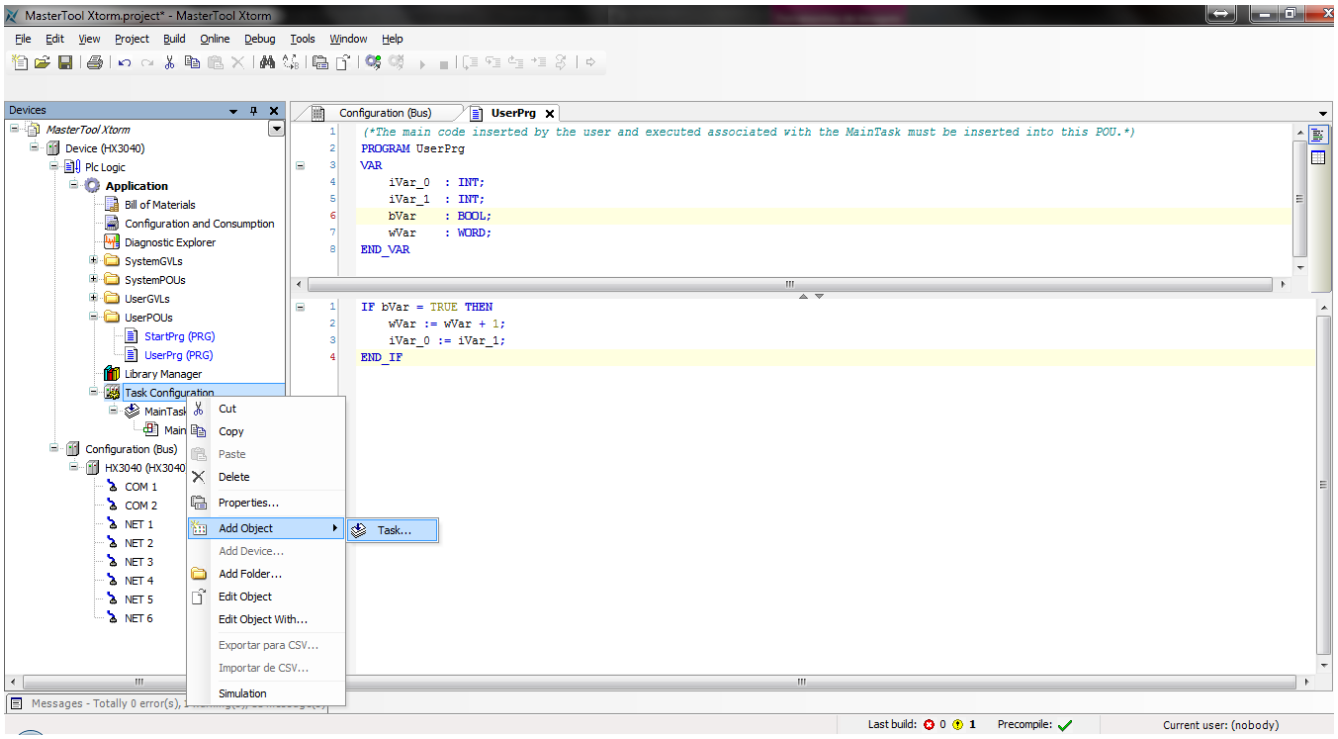


Figura 5-13. Criando uma Tarefa

Em seguida, surgirá uma tela para colocar o nome da tarefa. Após, clicar em Add para finalizar a criação da mesma.

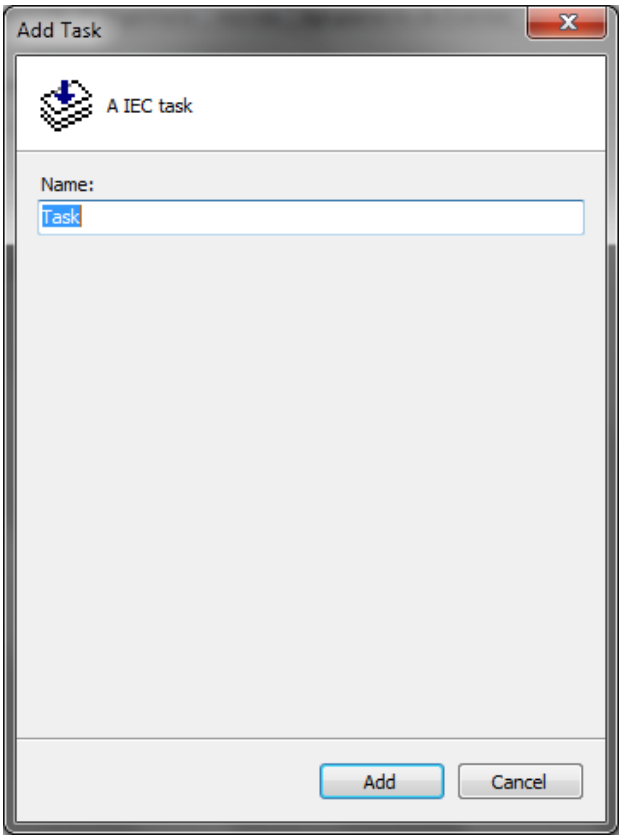


Figura 5-14. Nomeando a Tarefa

Configurando as Tarefas

Após abrir a tarefa, a janela de configuração surgirá para o usuário definir e classificar o funcionamento da mesma.

O campo “Priority (0..31)” estabelece a prioridade em que a tarefa será executada na aplicação, sendo o 0 (zero) mais prioritário. Por exemplo, a MainTask, criada em ambos os perfis de projeto, tem prioridade 13, ou seja, essa tarefa é considerada prioritária para o sistema.

O espaço “Type” define qual o tipo e o método para a tarefa ser executada, sendo que podem ser selecionados os seguintes itens:

- Cyclic: a tarefa é executada ciclicamente, ou seja, é chamada a cada intervalo de tempo configurado no campo ao lado. Ex: t#100 ms.
- Event: a tarefa é executada quando a variável do tipo BOOL, configurada no campo ao lado, recebe uma borda de subida, ou seja, a variável passe de FALSE para TRUE.
- External: a tarefa é executada quando uma interrupção externa ocorre, a qual é configurada no campo ao lado.
- Freewheeling: a tarefa sempre é executada, de acordo com a sua prioridade, ou seja, tarefas com maior prioridade são executadas primeiramente.
- Status: a tarefa é executada quando a variável do tipo BOOL, configurada no campo ao lado, for verdadeira.

Além dos campos mencionados acima, ainda deve ser configurado o Interval (apenas para tarefas cíclicas), ou seja, o intervalo de tempo em que a tarefa é chamada para executar (tempo máximo para a MainTask é igual a 750 ms e o tempo mínimo igual a 5 ms para projetos simples e 50 ms para projetos redundantes). Recomenda-se configurar o intervalo da tarefa de um modo que o tempo de ciclo (execução) seja, no máximo, 80% do intervalo.

O cão-de-guarda da UCP é configurado para evitar o travamento das tarefas de usuário.

O campo “Time”, define o tempo máximo permitido para a execução da tarefa. Caso a tarefa leve um tempo maior que o tempo de cão-de-guarda para ser executada, a aplicação irá para STOP e entrará em exceção por cão-de-guarda.

O campo “Sensivity” refere-se a quantas vezes o tempo do cão-de-guarda deverá ser atingido para que o diagnóstico bWatchdogReset seja ativado, por sua vez somente editável em um projeto com perfil personalizado. Caso o tempo de execução da tarefa atinja o valor do campo Sensivity multiplicado pelo campo Time, o diagnóstico também será indicado.

Deve-se atentar ao fato de que o cão-de-guarda da UCP não é utilizado para proteger a aplicação do usuário de picos no tempo de execução e sim de travamentos.

Visando proteger o sistema quanto a possíveis erros de configuração, o MasterTool Xtorm verifica em todas as tarefas cíclicas, durante a compilação, o cão-de-guarda (Cão-de-Guarda de Software) e os limites mínimo e máximo do intervalo da tarefa. É importante destacar que o usuário deverá ter cuidado ao alterar os valores pré-definidos pelos perfis de projeto, uma vez que alterações indevidas podem colocar em risco a execução do sistema. Então, recomenda-se utilizar os valores padrão, ou seja, manter o tempo de cão-de-guarda em 1000 ms.

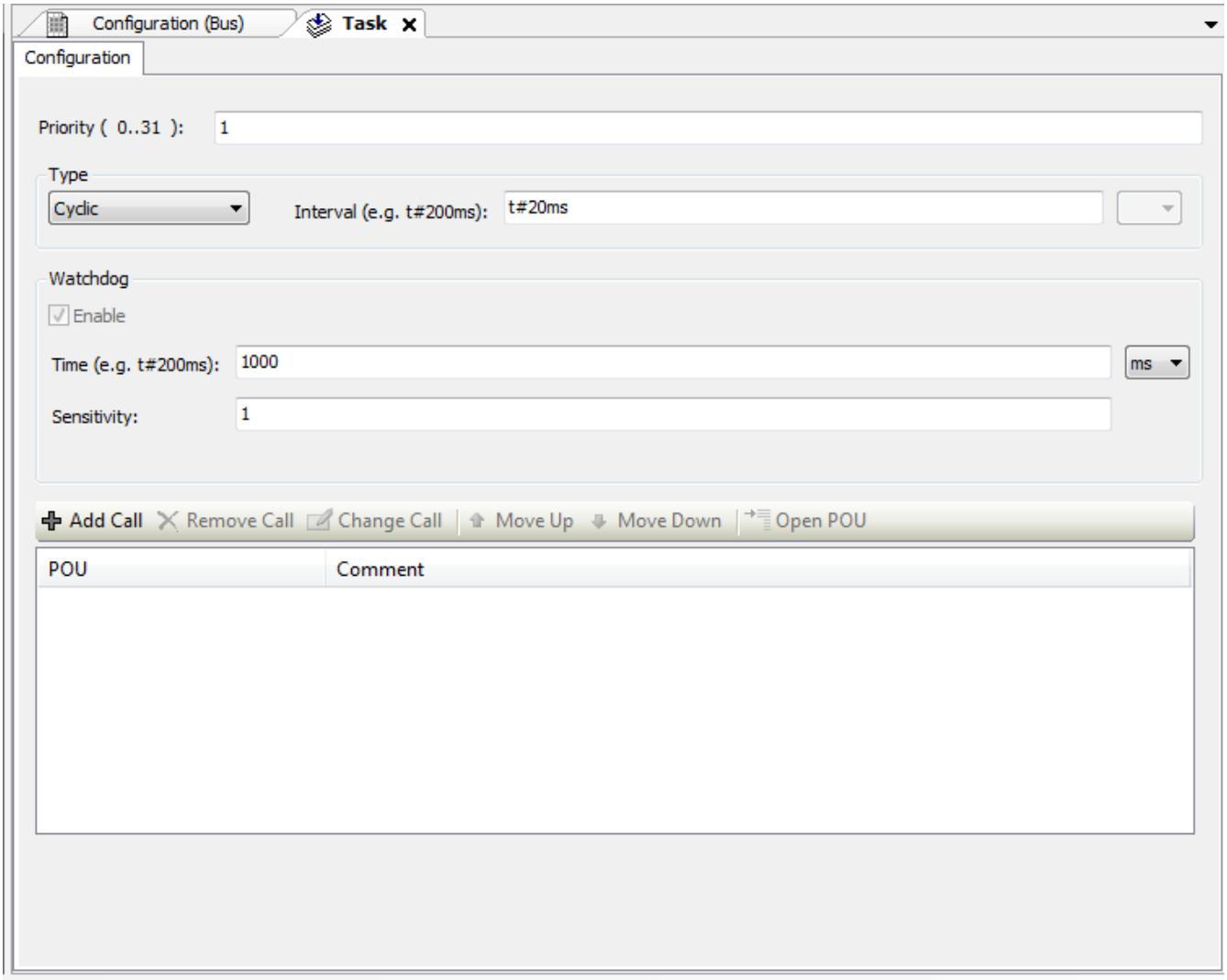


Figura 5-15. Configurando a Tarefa Criada

A Tabela 5-5 exibe as verificações realizadas pelo MasterTool Xtorm, sendo o campo Sensivity igual a 1. Para o Perfil Personalizado não é feita consistência no intervalo da tarefa e no tempo de cão-de-guarda.

Tarefa	Tipo	Intervalo Mínimo	Intervalo Máximo
MainTask	Cíclica	5 ms	750 ms
CyclicTask	Cíclica	5 ms	2147483 ms
TimeInterruptTask00	Cíclica	500 us	2147483 ms

Tabela 5-5. Configurações Máximas Permitidas

Vínculo de POUs com Tarefas

Como descrito anteriormente, para que uma POU seja executada na aplicação, ela deve estar associada a uma tarefa. Nos perfis de projeto (sem considerar o Personalizado), as POUs já estão associadas as suas respectivas tarefas. Porém, caso utilizado o perfil Personalizado ou novas POUs criadas, as mesmas devem ser vinculadas a tarefas.

Para associar uma POU criada, basta acessar a tarefa desejada, clicando duas vezes sobre a mesma na árvore de dispositivos, e clicar sobre "Add POU". Após, surgirá uma tela denominada "Input Assistant", ou seja, Assistente de Entrada na qual deverá ser selecionada a POU desejada, conforme mostra a Figura 5-16:

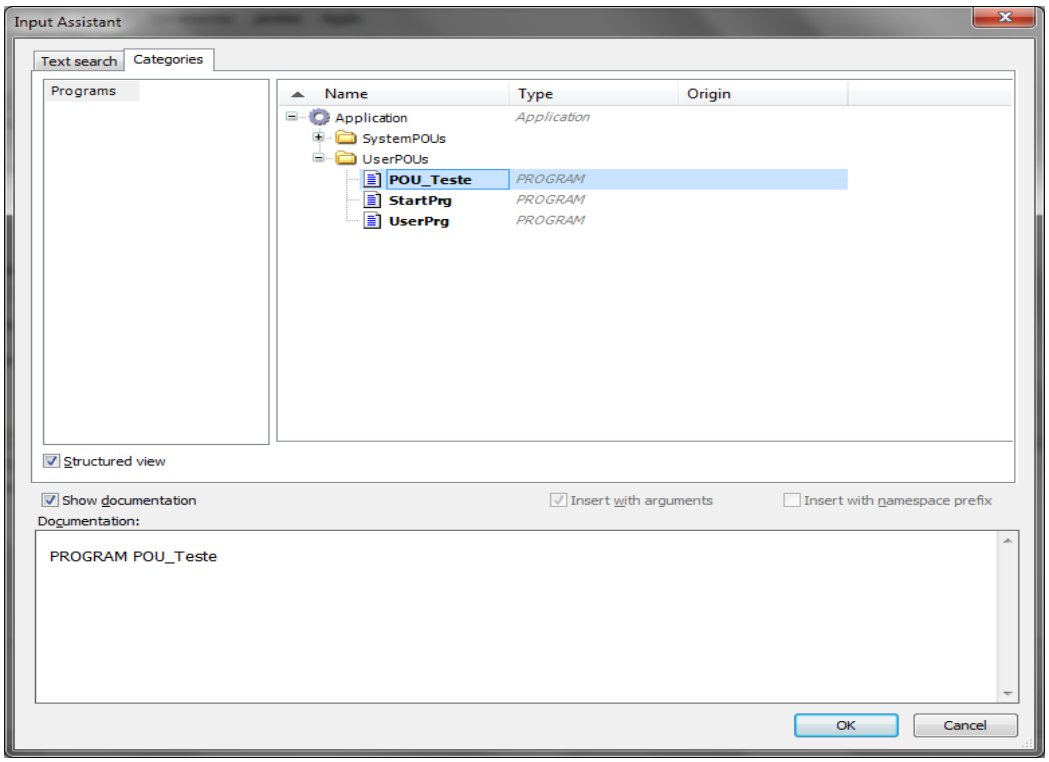


Figura 5-16. Vinculando POUs às Tarefas

Outra opção é a chamada de uma POU dentro de uma POU já instanciada no programa, segue exemplo onde a POU_Teste será chamada dentro da área destinada a código da POU UserPrg:

```
POU_Teste();
```

Número Máximo de Tarefas

O número máximo de tarefas que o usuário poderá criar somente está definido para o perfil Personalizado, ou seja, o único que tem essa permissão. Os demais já têm as suas tarefas criadas e configuradas.

A Tabela 5-6 descreve a quantidade máxima de tarefas IEC por UCP e perfil de projeto, sendo que as instâncias de protocolo também são consideradas tarefas de comunicação pela UCP.

	Tipo de Tarefa	HX3040	
		UTR	Personalizado
Configuration Task (Tarefa WHSB)	Cíclica	1*	1
Tarefas de Usuário	Cíclica	1*	15*
	Disparada por Evento	0	15*
	Disp. Evento Externo	0	15*
	Contínua	0	15*
	Disparada por Estado	0	15*
NETs – Instâncias Cliente ou Servidor	Cíclica	4	
COM (n) – Instâncias Mestre ou Escravo	Cíclica	1	
TOTAL		16	32

Tabela 5-6. Número Máximo de Tarefas IEC

Notas:

Valores: Os números definidos para cada tipo de tarefa representam os valores máximos permitidos.

Tarefa WHSB: A tarefa WHSB que é uma tarefa de sistema deve ser considerada para que não seja ultrapassado o valor total. Para perfil UTR utilizando redundância pode se optar por criar também a tarefa ProtTask, totalizando assim até 2 tarefas cíclicas.

NETs: Instâncias Cliente ou Servidor: O valor máximo definido considera todas as interfaces Ethernet do sistema, ou seja, inclui os módulos de expansão, quando estes são aplicados. Exemplos para esse tipo de tarefa são as instâncias do Protocolo MODBUS.

COM (n): Instâncias Mestre ou Escravo: O "n" representa o número da interface serial, ou seja, mesmo com módulos de expansão, o valor da tabela será o máximo por interface. Exemplos para esse tipo de tarefa são as instâncias do Protocolo MODBUS.

Total: O valor total não representa a soma de todas as tarefas por perfil, mas o valor máximo permitido por UCP. Então, o usuário poderá criar vários tipos de tarefas, desde que o número estabelecido para cada uma e o valor total, não sejam ultrapassados.

Configurando a UCP

A configuração da UCP Hadron Xtorm baseia-se em configurar os parâmetros da UCP sobre o Cão de guarda, troca a quente, sincronismo de tempo, pontos internos, conversão de engenharia, alarmes e agrupamento de eventos.

O usuário deverá dar dois cliques na UCP Hadron Xtorm, localizada na árvore de dispositivos, conforme mostra a Figura 5-17, e configurar os campos conforme descrito no capítulo Configuração da UCP.

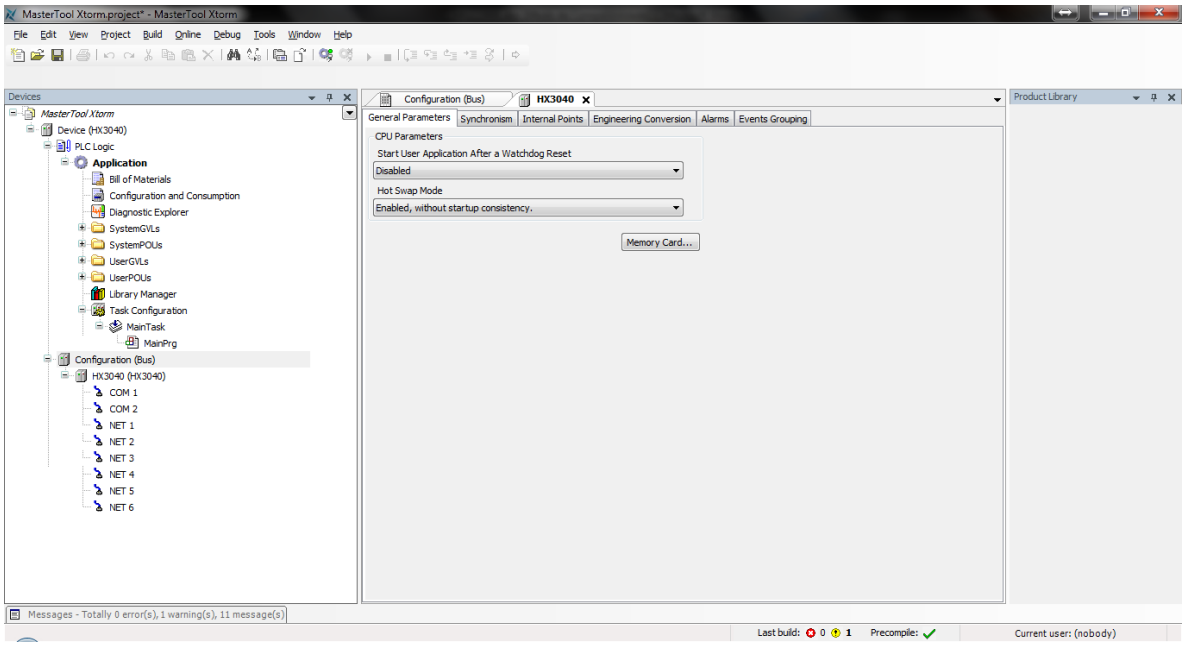


Figura 5-17. Configuração da UCP

Além disso, para que seja possível a comunicação entre a UCP e o MasterTool Xtorm, a interface Ethernet NET 1 deverá ser configurada, conforme o capítulo Configuração das Interfaces Ethernet– NET 1.Clicando duas vezes sobre o ícone NET 1 da UCP, na árvore de dispositivos, surgirá uma nova aba para configuração da rede de comunicação em que o módulo está conectado.

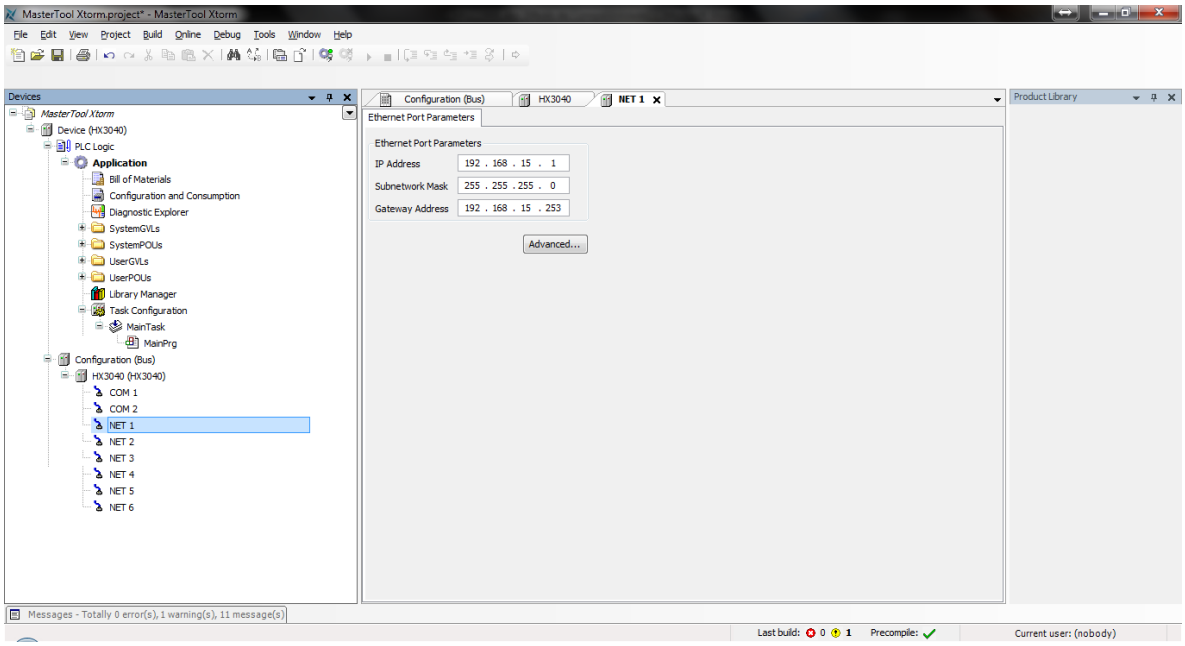


Figura 5-18. Configurando a Porta de Comunicação da UCP

Caso a UCP com o IP configurado não seja encontrada na rede ou a UCP que está atualmente ativa tenha um IP diferente da ativa configurada no projeto, uma mensagem surgirá na tela durante o Login, solicitando ao usuário a possibilidade de substituir o IP antigo pelo configurado (opção Sim) ou Não para não enviar o projeto.

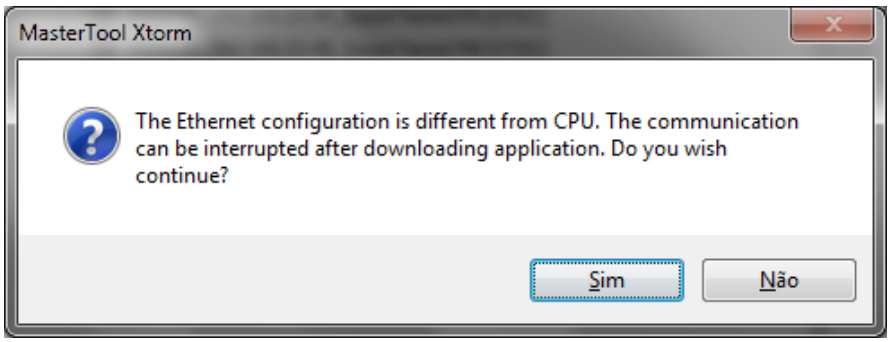


Figura 5-19. Aviso de configuração IP

Bibliotecas

Existem diversos recursos da ferramenta de programação que estão disponíveis através de bibliotecas. Sendo assim, os mesmos devem ser inseridos no projeto para que a sua utilização seja possível. O procedimento de inserção é bastante simples, sendo necessário selecionar o item “Library Manager”, disponível na treeview do menu Dispositivo e selecionar “Add Library”, conforme mostra a Figura 5-20:

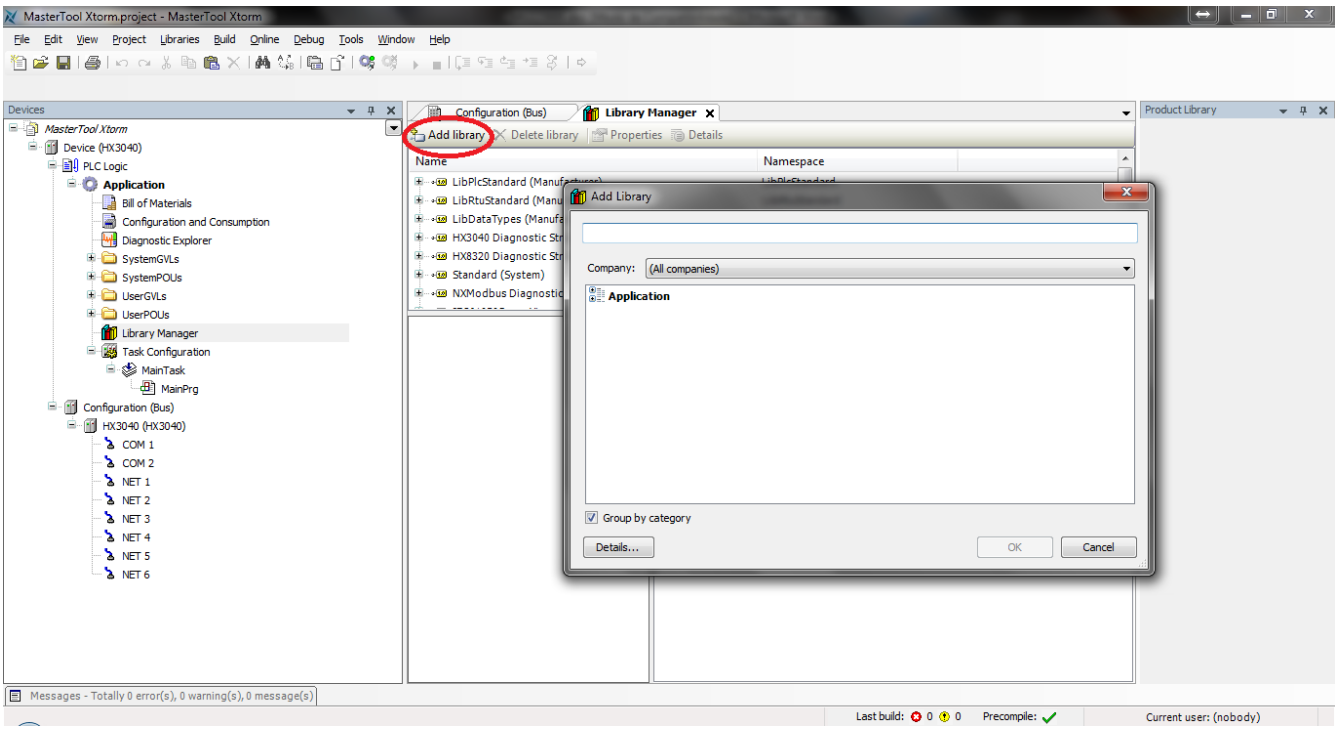


Figura 5-20. Inclusão de uma Biblioteca no Projeto

Na sequência, deve-se selecionar a biblioteca desejada para inclusão no projeto, pressionando em seguida, o botão OK.

Inserindo uma Instância de Protocolo

A própria UCP da Série Hadron Xtorm, conforme o capítulo Características Gerais, disponibiliza protocolos, como o MODBUS. Basta adicionar e configurar (conforme o capítulo Configuração de Protocolos) a instância do protocolo desejado na interface de comunicação.

Abaixo, são descritos dois casos de inserção do protocolo MODBUS, sendo um na interface serial e outro na interface Ethernet.

MODBUS RTU

O primeiro passo para configurar o MODBUS RTU, em modo escravo, é incluir a instância na COM desejada (nesse caso, COM 1). Clicar com o botão direito sobre a COM e selecionar “Add Device...”, conforme mostra a Figura 5-21:

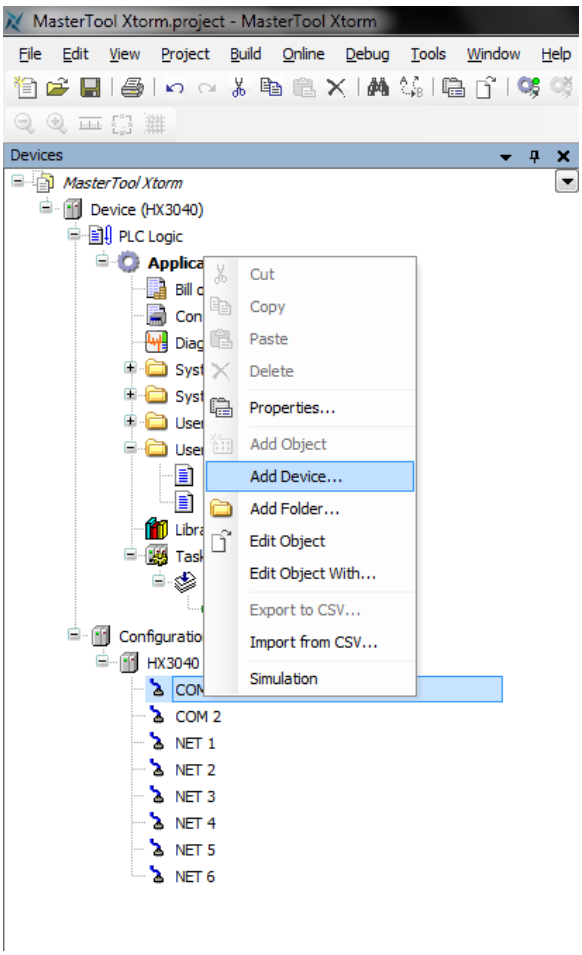


Figura 5-21. Adicionando a Instância

Após, surgirão na tela os protocolos disponíveis ao usuário. Nesse caso, selecionar o “MODBUS Slave” expandir a árvore de opções, selecionar o dispositivo e clicar em Add Device, conforme mostra a Figura 5-22:

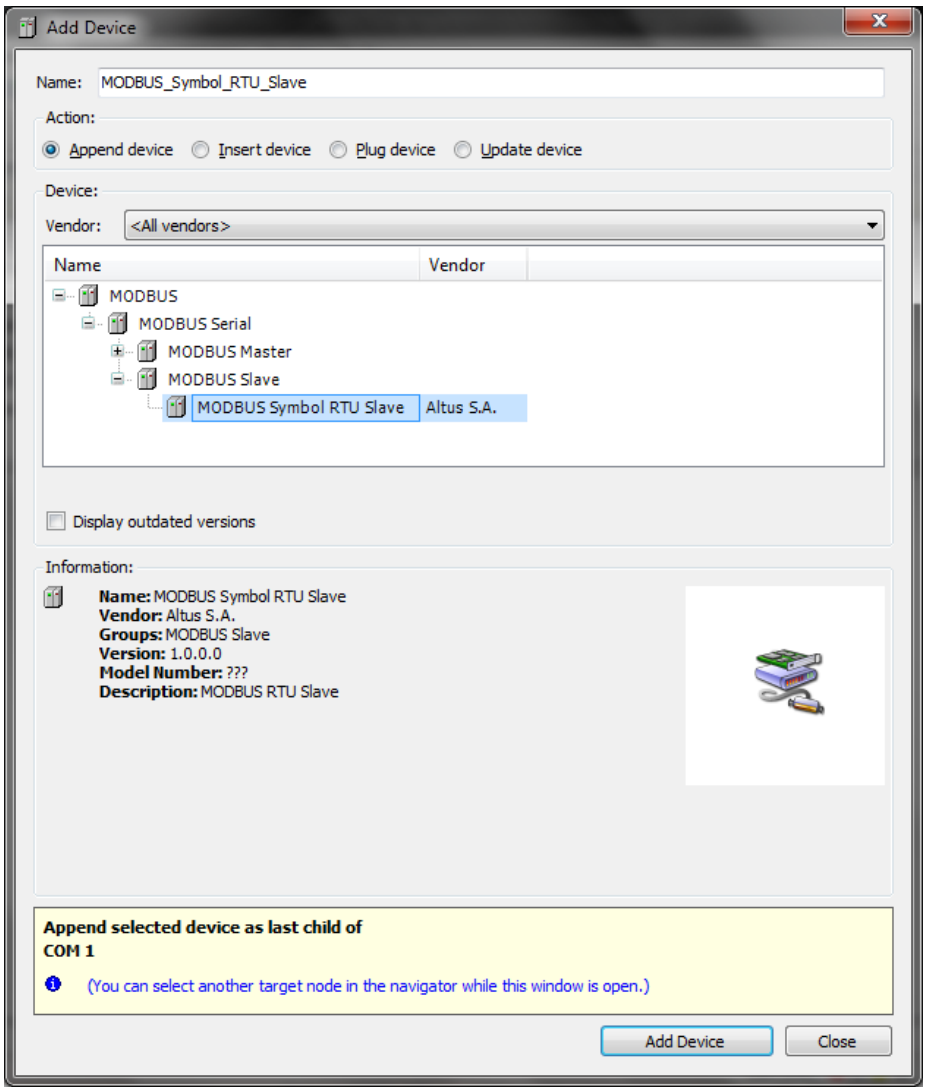


Figura 5-22. Selecionando o Protocolo

MODBUS Ethernet

O primeiro passo para configurar o MODBUS Ethernet, em modo cliente, é incluir a instância na NET desejada (nesse caso NET 1, a UCP HX3040 possui seis interfaces Ethernet). Clicar com o botão direito sobre a NET e selecionar “Add Device...”, conforme mostra a Figura 5-23:

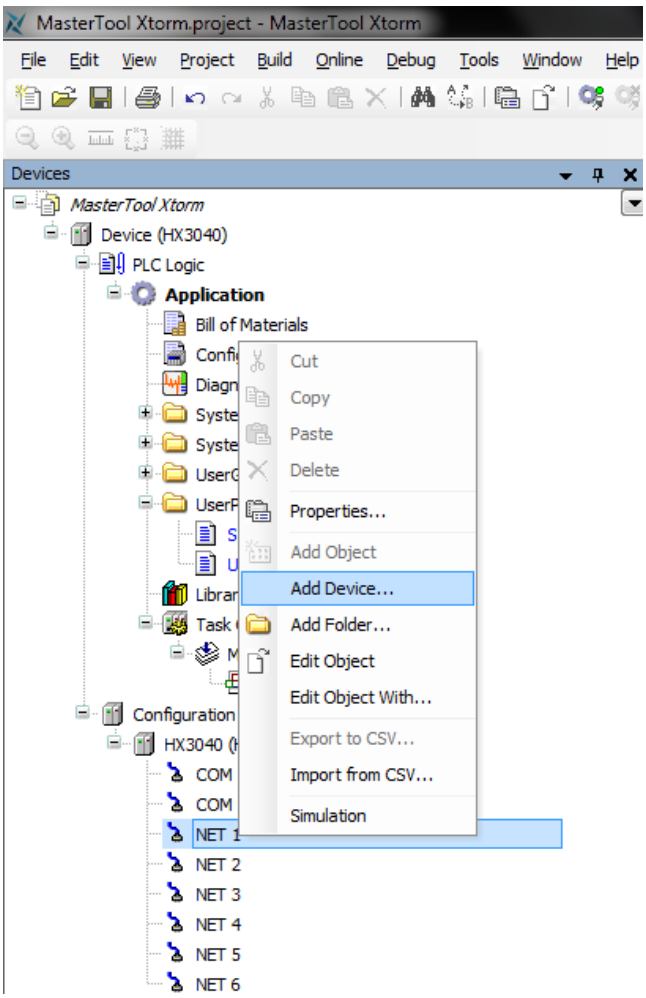


Figura 5-23. Adicionando a Instância

Após, surgirá na tela os protocolos disponíveis ao usuário. Nesse caso, selecionar “MODBUS > MODBUS Ethernet > MODBUS Client > MODBUS Symbol Client” e clicar em Add Device, conforme mostra a Figura 5-24:

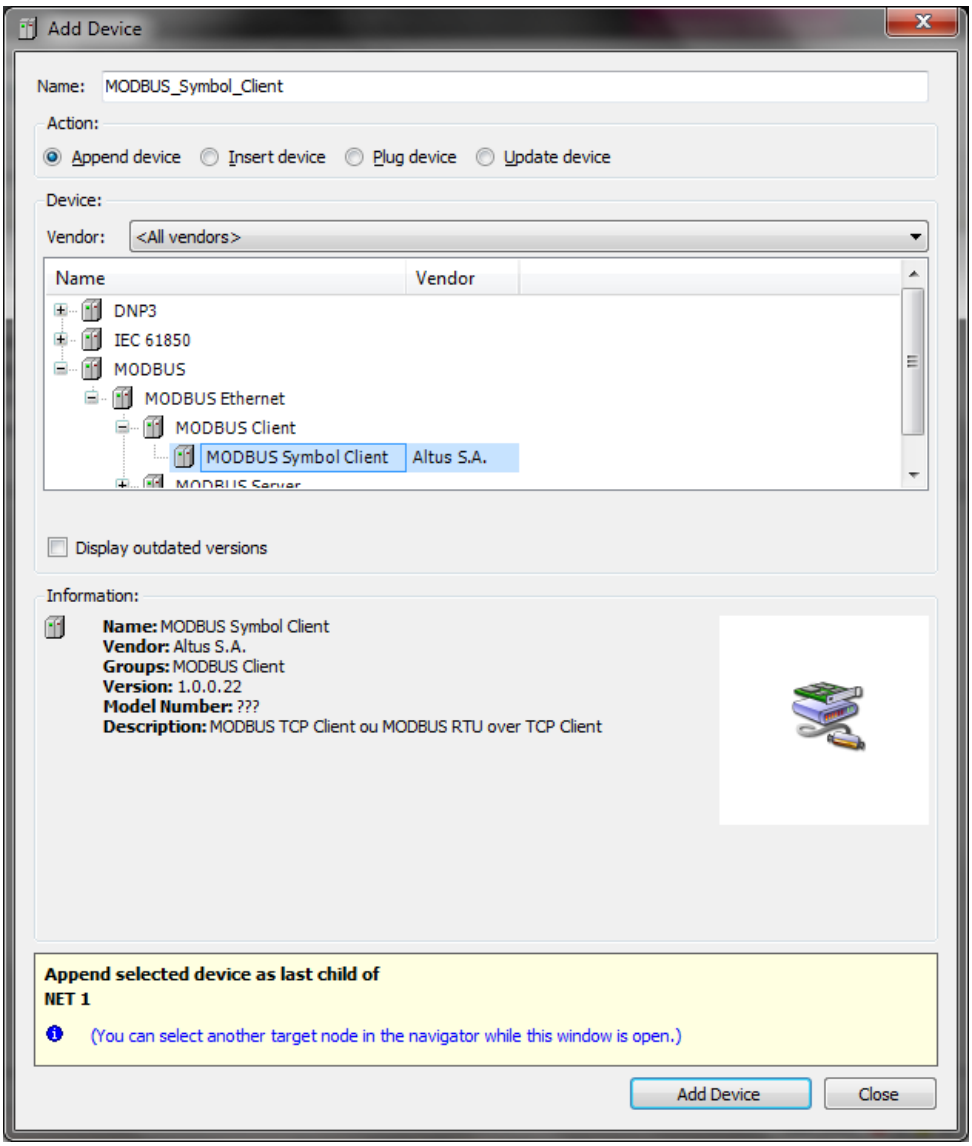


Figura 5-24. Selecionando o Protocolo

Servidor DNP3

O primeiro passo para configurar o DNP3, em modo servidor, é incluir a instância na NET desejada (nesse caso NET 1, a UCP HX3040 possui seis interfaces Ethernet). Clicar com o botão direito sobre a NET e selecionar “Add Device...”, conforme mostra a Figura 5-25:

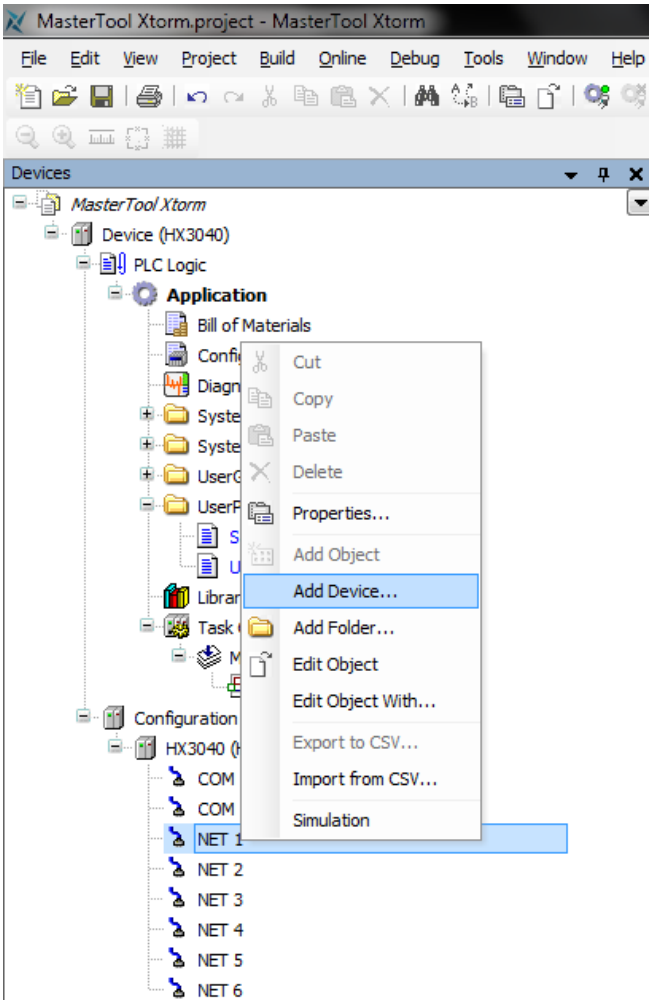


Figura 5-25. Adicionando a Instância Servidora DNP3

Após, surgirá na tela os protocolos disponíveis ao usuário. Nesse caso, selecionar o “DNP3 > DNP3Ethernet > DNP3 Server > DNP3 Server” e clicar em Add Device, conforme mostra a Figura 5-26:

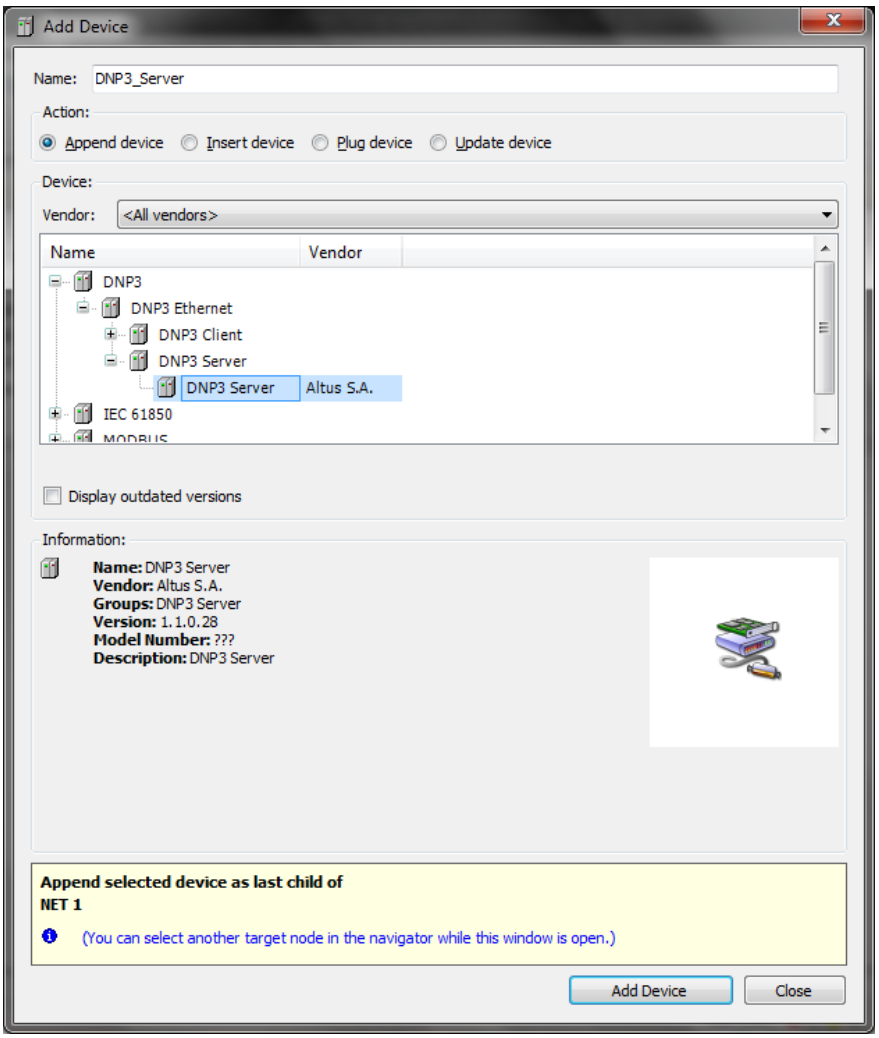


Figura 5-26. Selecionando o Protocolo Servidor DNP3

Localizando a Rede

Como existe a possibilidade de mais UCPs estarem conectadas à rede, o usuário deve localizar todas as unidades de comunicação e selecionar a desejada.

Inicialmente, será necessário acessar a opção Device, na árvore de dispositivos, clicando duas vezes. Na aba “Communication Settings”, selecionar o Gateway, ou, caso nenhum Gateway exista ou o usuário queira adicionar um novo Gateway, clicar no botão “Add Gateway”, configurando seu IP na janela que será aberta. Para mapear os dispositivos presentes na rede, deve-se então clicar no botão “Scan Network".

Então, deve-se aguardar enquanto o software MasterTool Xtorm procura e apresenta as UCPs disponíveis na rede.

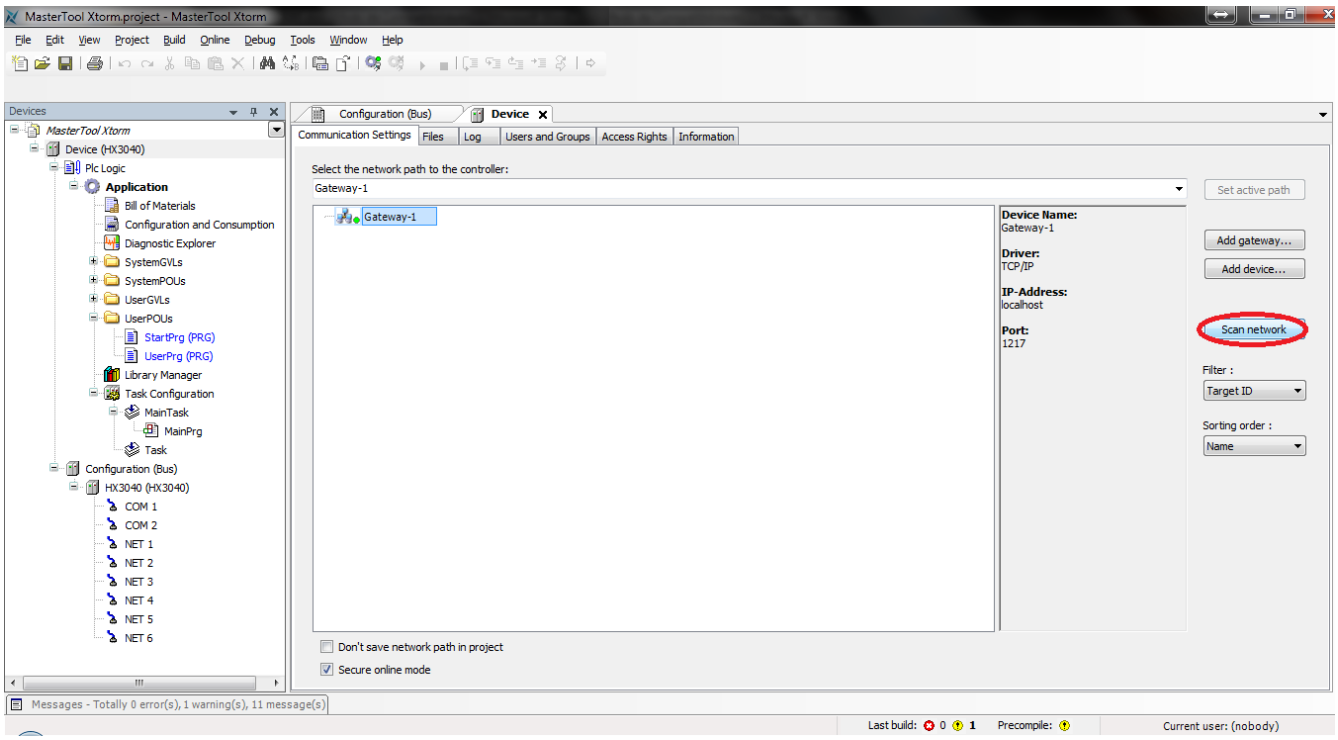


Figura 5-27. Localizando a UCP

Após, basta selecionar a UCP desejada e clicar na opção “Set Active Path”, fazendo com que ela fique ativa e o configurador saiba com quem deve comunicar e enviar o projeto.

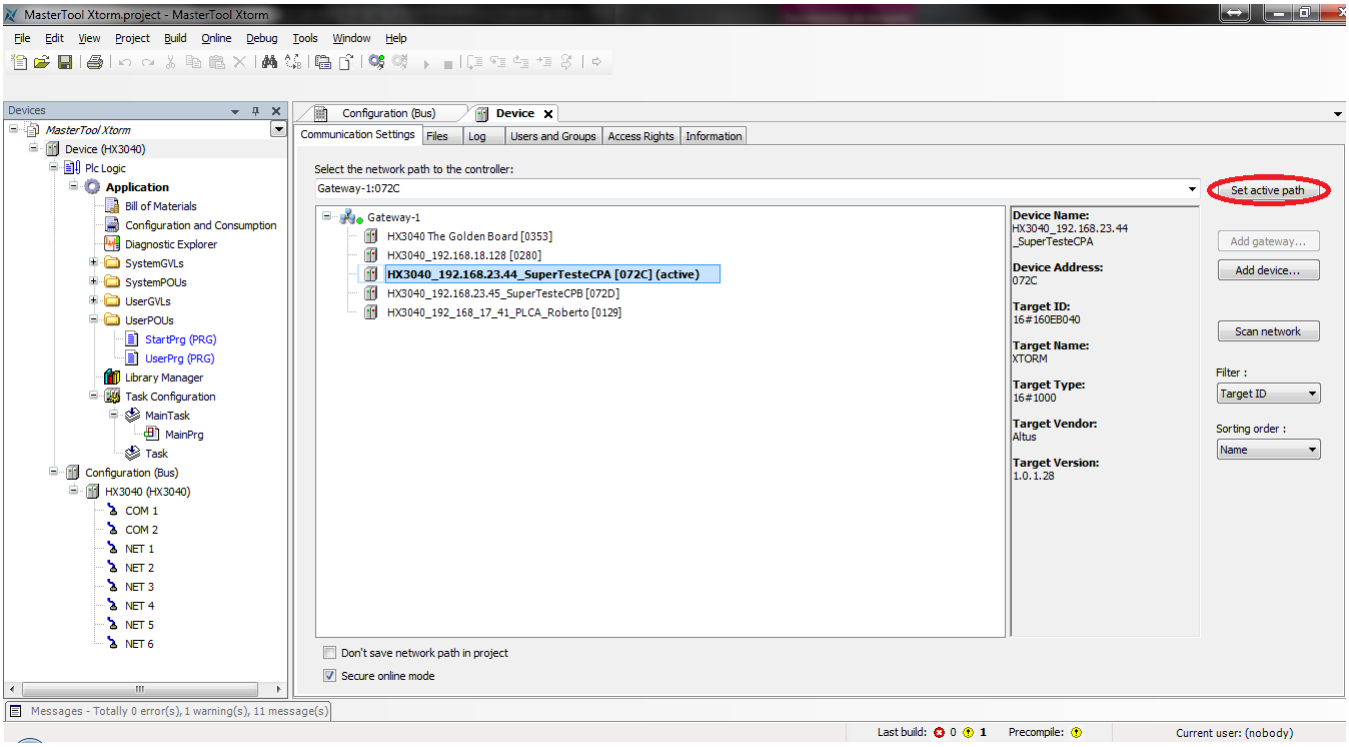


Figura 5-28. Ativando a UCP

Caso seja necessário, o usuário pode alterar o nome padrão do dispositivo que é exibido. Para isso, deve clicar com o botão direito do mouse sobre o dispositivo desejado e selecionar a opção “Change

Device Name". Após uma mudança de nome, o dispositivo não voltará ao nome padrão em nenhuma circunstância.

Compilando um Projeto

A fim de realizar a verificação da aplicação criada, o usuário deve executar a compilação do projeto. Essa é a forma mais eficaz de se encontrar problemas ou receber avisos sobre alguns equívocos cometidos durante a configuração do produto e edição da aplicação. Para executar tal procedimento, basta acessar o menu Build e clicar em "Generate Code" ,conforme mostra a Figura 5-29.

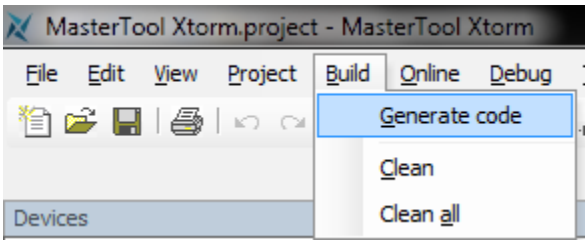


Figura 5-29. Compilando o Projeto

Depois do tempo de processamento, o qual vai variar de acordo com o tamanho da aplicação do usuário, os erros e mensagens de alerta, caso sejam necessários, serão mostrados abaixo, conforme mostra a Figura 5-30.

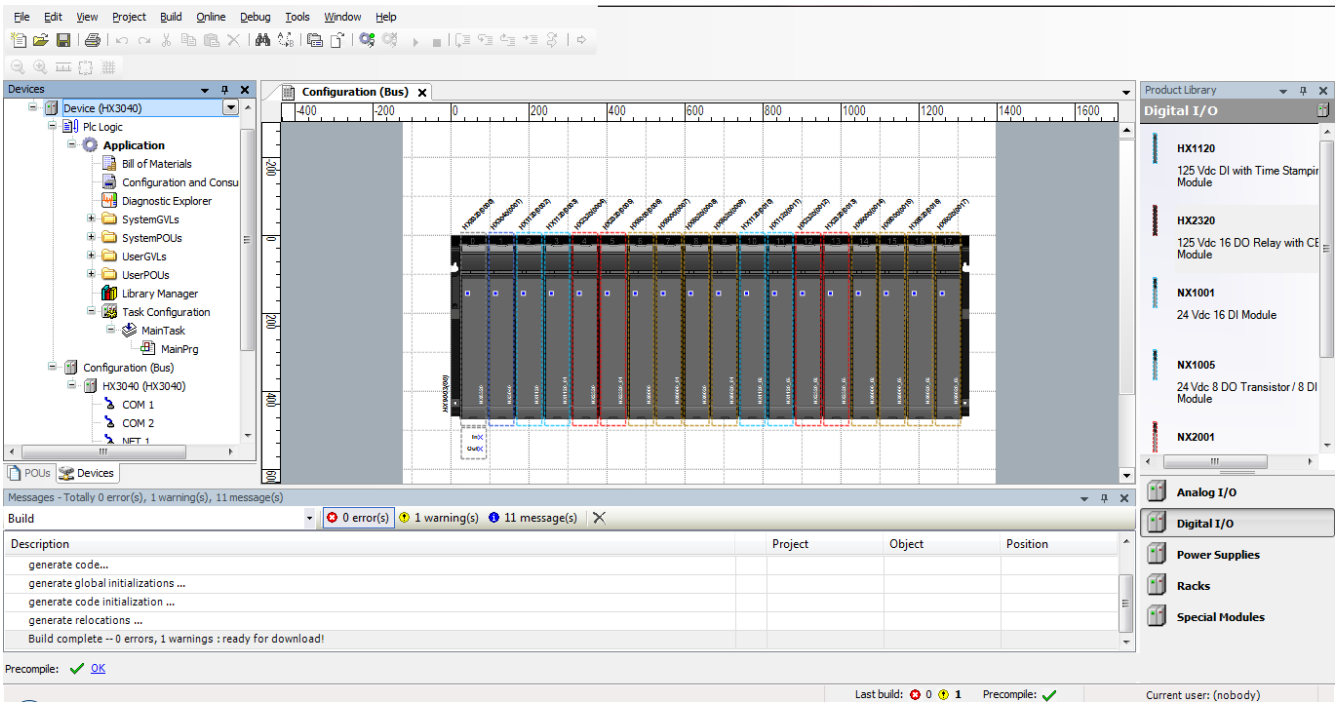


Figura 5-30. Mensagens da Compilação

Obs.: Caso os erros e mensagens não estejam visíveis na tela, basta acessar o menu Visualizar e clicar em "Messages", conforme mostra a Figura 5-31.

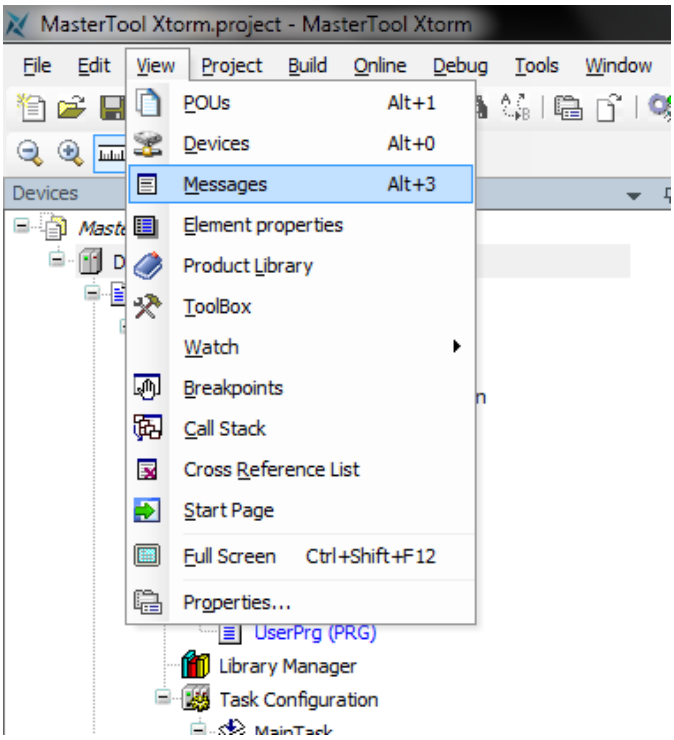


Figura 5-31. Incluindo as Mensagens na Tela

Login

Após compilar a aplicação e corrigir todos os erros encontrados, é o momento de enviar o projeto para a UCP. Para que isso seja possível, basta efetuar a operação de Login no software MasterTool Xtorm. Essa operação pode levar alguns segundos, dependendo do tamanho do arquivo gerado. Para executar o Login, basta acessar o menu Online e clicar na opção “Login”, conforme mostra o exemplo da Figura 5-32.

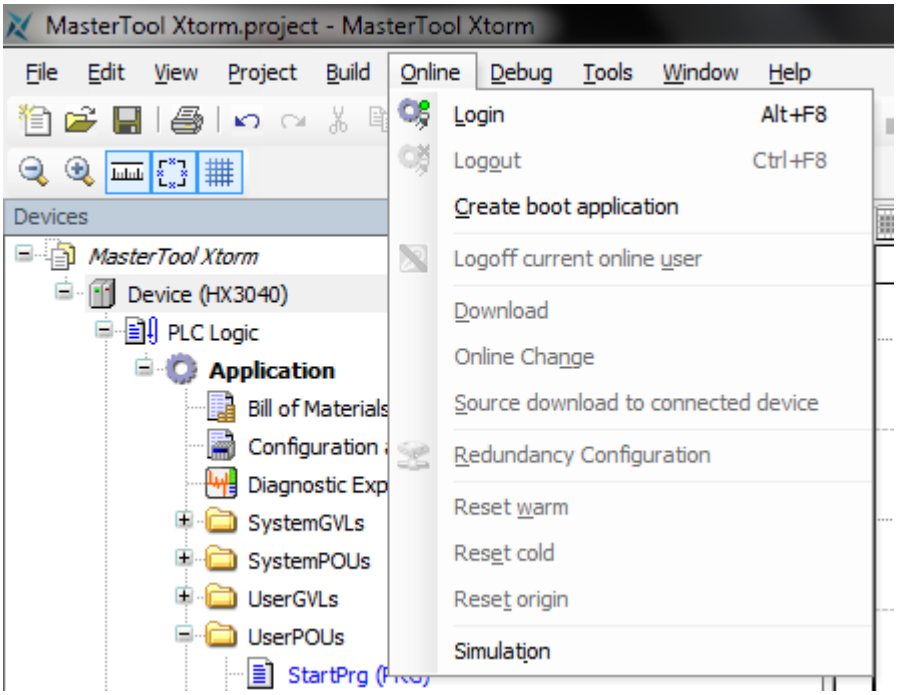


Figura 5-32. Enviando o Projeto para a UCP

Após a execução do comando, poderão surgir algumas mensagens de interface com o usuário, as quais são apresentadas devido a diferença entre um projeto antigo e o que está sendo enviado ou, simplesmente, alteração no valor de alguma variável.

A Figura 5-33 mostra a mensagem que o MasterTool Xtorm irá apresentar em caso do novo projeto, que está sendo enviado seja diferente de um projeto já existente dentro da UCP. As opções disponíveis são as seguintes:

- Login with Online Change: executar o Login e enviar o novo projeto sem que a aplicação atual da UCP seja parada (ver item Modo Run), atualizando as alterações quando um novo ciclo é executado.
- Login with Download: executar o Login e enviar o novo projeto com a UCP parada (ver item Modo Stop). Quando a aplicação for iniciada, as atualizações já terão sido realizadas.
- Login Without Any Change: executa o Login sem enviar o novo projeto.

ATENÇÃO: Quando o Login com alteração online for executado, a aplicação não ficará salva na memória de programa. É necessário executar o comando "Criar Aplicação de Inicialização" no menu Comunicação, sem efetuar o Logout, para que a aplicação seja gravada na memória de programa.

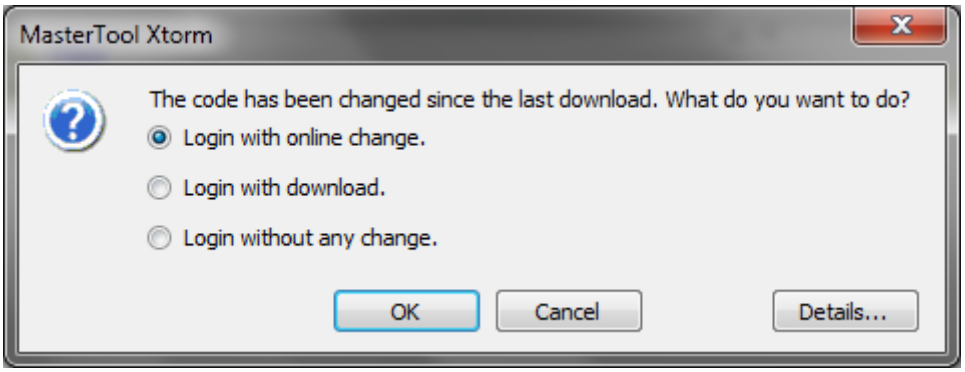


Figura 5-33. Atualização do Projeto na UCP

A Figura 5-34 mostra a mensagem que o MasterTool Xtorm exibe quando somente foram realizadas alterações em variáveis da aplicação, fazendo com que não seja possível enviar o novo projeto e o mesmo ser atualizado em um novo ciclo da UCP, a qual está em execução (ver item Modo Run). Dessa forma, o MasterTool Xtorm solicita ao usuário se o Login deve ser executado como download e a UCP parada (ver item Modo Stop) ou a operação deve ser cancelada.

Obs.: O botão "Details..." apresenta as modificações realizadas na aplicação.

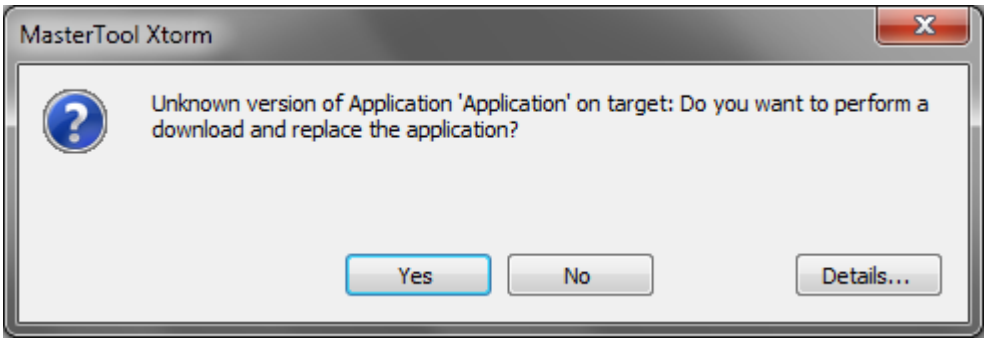


Figura 5-34. Alteração de Variáveis

Caso seja o primeiro envio de aplicação para a UCP, a mensagem da Figura 5-35 irá surgir na tela do MasterTool Xtorm.

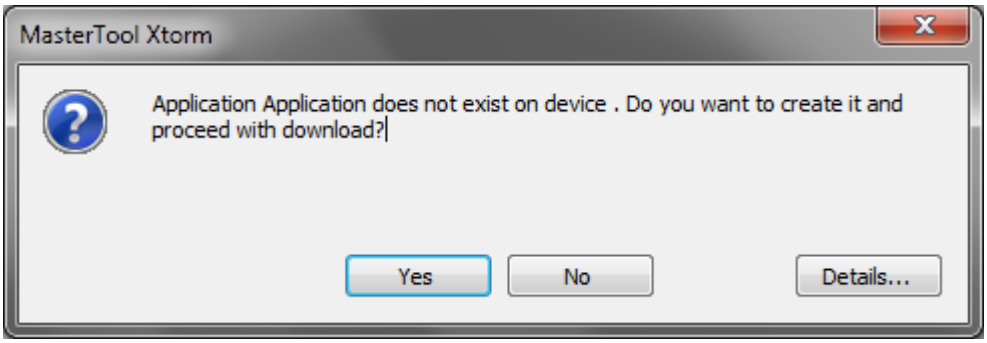


Figura 5-35. Primeiro Envio de Aplicação

Modo Run

Logo após enviar o projeto para a UCP, a aplicação não será executada imediatamente (somente se for realizada uma alteração online), antes que seja selecionado um novo comando “Iniciar”. Essa função possibilita ao usuário controlar a execução da aplicação enviada para a UCP. Além disso, permite que valores iniciais sejam pré-configurados, para que, no primeiro ciclo já sejam atualizados na UCP.

Para selecionar tal funcionalidade, basta acessar o menu Debug e clicar em “Start”, conforme mostra a Figura 5-36.

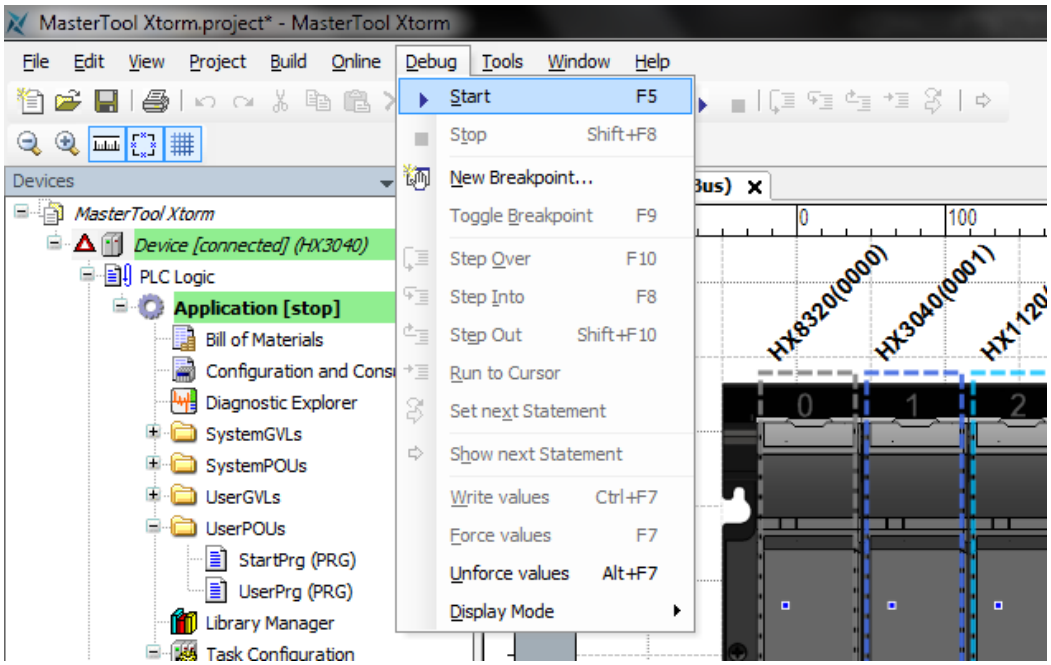


Figura 5-36. Iniciando a Aplicação

A Figura 5-37 mostra a aplicação em execução. Caso seja selecionada a aba de uma POU, as variáveis criadas serão listadas em uma janela de monitoração, na qual valores podem ser forçados e visualizados pelo usuário. Caso as variáveis sejam forçadas através do comando F7 do teclado, a UCP indicará essa condição no visor gráfico. Maiores detalhes consultar capítulo Manutenção – Visor Gráfico.

É importante ressaltar que, quando utilizados os protocolos MODBUS RTU Escravo e MODBUS Ethernet Servidor e a opção “Somente Leitura” das relações configuradas não estiver selecionada, o comando de escrita forçada (F7) deve ser realizado sobre as variáveis disponíveis na janela de monitoração, pois o comando de escrita (CTRL + F7) deixa as variáveis serem sobrescritas quando realizadas novas leituras.

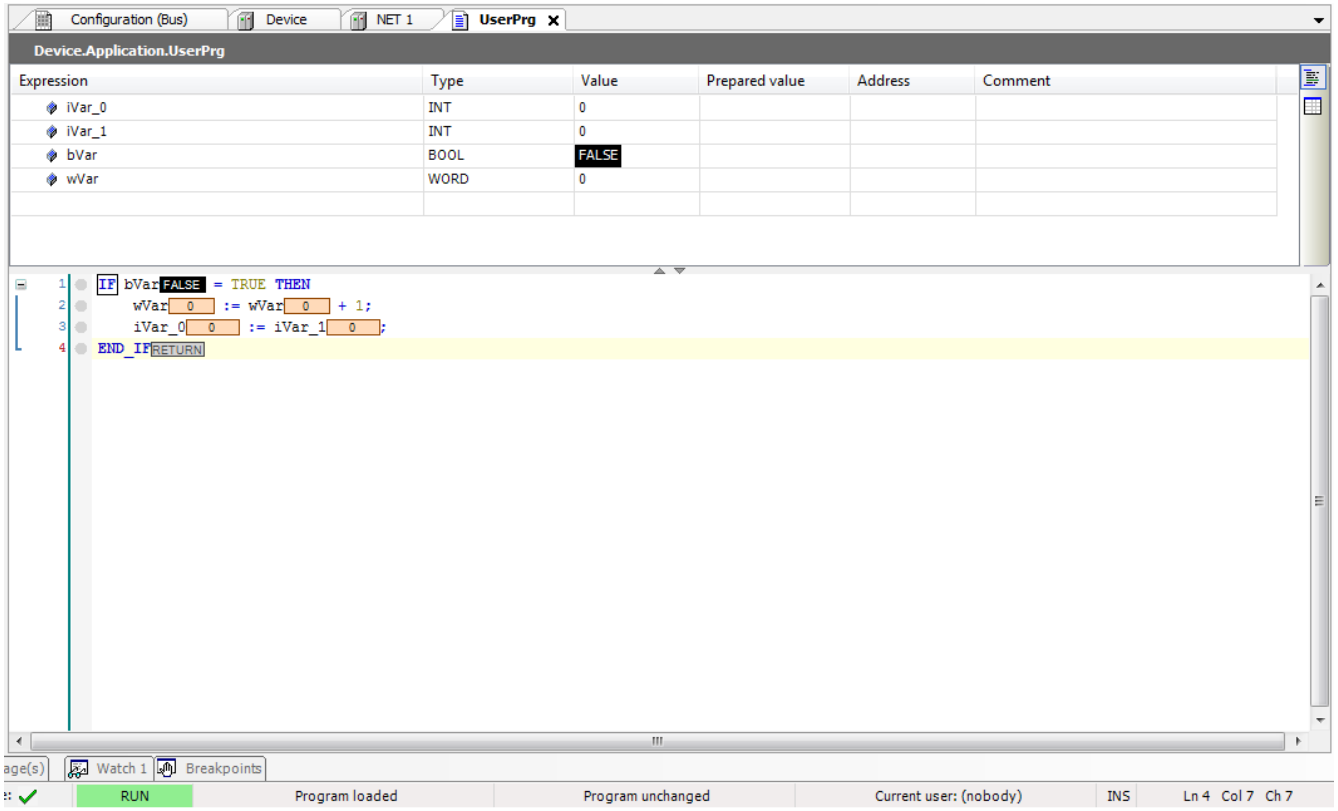


Figura 5-37. Programa em Execução

Caso a UCP é inicializada como uma aplicação já gravada internamente, ela automaticamente entra em Modo Run, sem a necessidade de realizar o comando via MasterTool Xtorm.

Modo Stop

Para interromper a execução da UCP, sem que o software MasterTool Xtorm perca a conexão com a mesma, o usuário deve selecionar a opção Parar, disponível no menu Debug, “Stop”, conforme mostra a Figura 5-38.

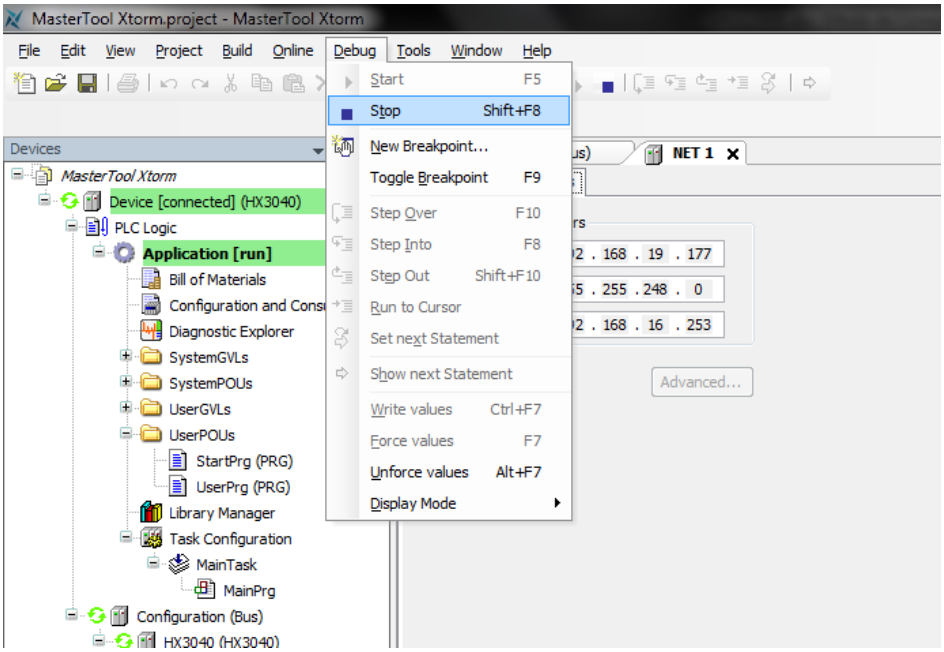


Figura 5-38. Parando a Aplicação

Caso a UCP seja inicializada sem aplicação gravada, ela automaticamente entra em Modo Stop, assim como quando ocorre uma exceção de software.

Monitoração, Escrita e Forçamento de Variáveis

Após efetuar um Login em uma UCP, o usuário poderá monitorar variáveis do projeto de 3 formas distintas:

- Diretamente na POU (na área de declaração de variáveis ou na área do programa)
- Através da janela de monitoração (menu View -> Watch)
- Utilizando a funcionalidade Trace, a qual realiza a gravação e exibição dos valores das variáveis internamente na UCP.

ATENÇÃO: No modo online há uma limitação de 25000 entradas de variáveis monitoráveis em POUs editadas com o editor ST, o usuário será alertado quando o limite for ultrapassado com um erro de compilação.

Além disto, também é possível escrever ou forçar valores em variáveis do projeto.

O comando de escrita (CTRL + F7) escreve um valor em uma variável, podendo este valor ser sobrescrito pelas instruções executadas na aplicação no próximo ciclo de execução.

Já um comando de escrita forçada (F7) irá escrever um valor na variável, sem permitir que este valor seja alterado até que sejam liberadas as variáveis forçadas.

ATENÇÃO: O forçamento de variáveis pode ser realizado somente em modo Online na UCP. Variáveis de diagnóstico não podem ser forçadas, apenas escritas, pois diagnósticos são providos pela UCP e devem poder ser sobrescritos pela mesma.

Quando for efetuada uma escrita forçada em uma variável redundante da UCP Ativa, a MainTask da aplicação sofrerá um impacto em seu tempo de execução, tanto na UCP Ativa, quando na UCP Reserva. Isto porque as duas UCPs irão trocar a cada ciclo informações sobre as variáveis forçadas. Portanto, quando for forçar variáveis em um sistema redundante, deve-se considerar o acréscimo que pode ter a execução da tarefa. A Tabela 5-7 exemplifica em quanto será acrescida, em média, a execução da MainTask quando isto ocorrer:

	UCP Ativo			UCP Reserva		
Tempo de Execução	50 ms	100 ms	200 ms	50 ms	100 ms	200 ms
Acréscimo com 10 forçamentos	2,4 %	2,2 %	1,7 %	4,0 %	3,4 %	2,0 %
Acréscimo com 50 forçamentos	12,0 %	9,2 %	6,0 %	18,0 %	12,0 %	8,0 %
Acréscimo com 128 forçamentos	26,0 %	21,0 %	16,0 %	56,0 %	34,0 %	22,5 %

Tabela 5-7. Influência do Forçamento de Variáveis em uma UCP Redundante

Para mais informações sobre forçamentos de variáveis no sistema redundante, consulte Sincronização de Dados Redundantes.

ATENÇÃO: Quando uma UCP está com variáveis forçadas e é desenergizada, na próxima inicialização as variáveis perderão o forçamento. O limite de forçamentos para a UCP Hadron Xtorm é de 128 variáveis, independente do modelo de UCP ou configuração utilizados.

Variáveis utilizadas em mais de uma origem

Uma origem define-se com um ponto de um módulo de E/S ou de um driver de comunicação que escreve em uma variável.

Quando um projeto é compilado, o Mastertool Xtorm verifica todos os pontos e informa se alguma variável está sendo utilizada em mais de uma origem. Essa verificação é realizada somente para campos informados na Tabela 5-8.

Módulo / Driver	Funcionalidade	Campo / Origem
Campos Validados		
HX3040	Agrupamento de Eventos	Nome da Variável Alarme de Atraso Excessivo
Dispositivo MODBUS do Cliente MODBUS	Mapeamentos	Nome da Variável Qualidade
Dispositivo MODBUS do Mestre MODBUS	Mapeamentos	Nome da Variável Qualidade
Outstation do Cliente DNP3	Mapeamentos	Nome da Variável Qualidade
Módulo de E/S	Bus: Mapeamento de E/S	Variável

Tabela 5-8. Variáveis utilizadas em mais de uma origem

Logout

Caso a opção do usuário seja finalizar a comunicação com a UCP, deve ser utilizado comando “Logout”, localizado no menu Online, “Logout”, conforme mostra a Figura 5-39.

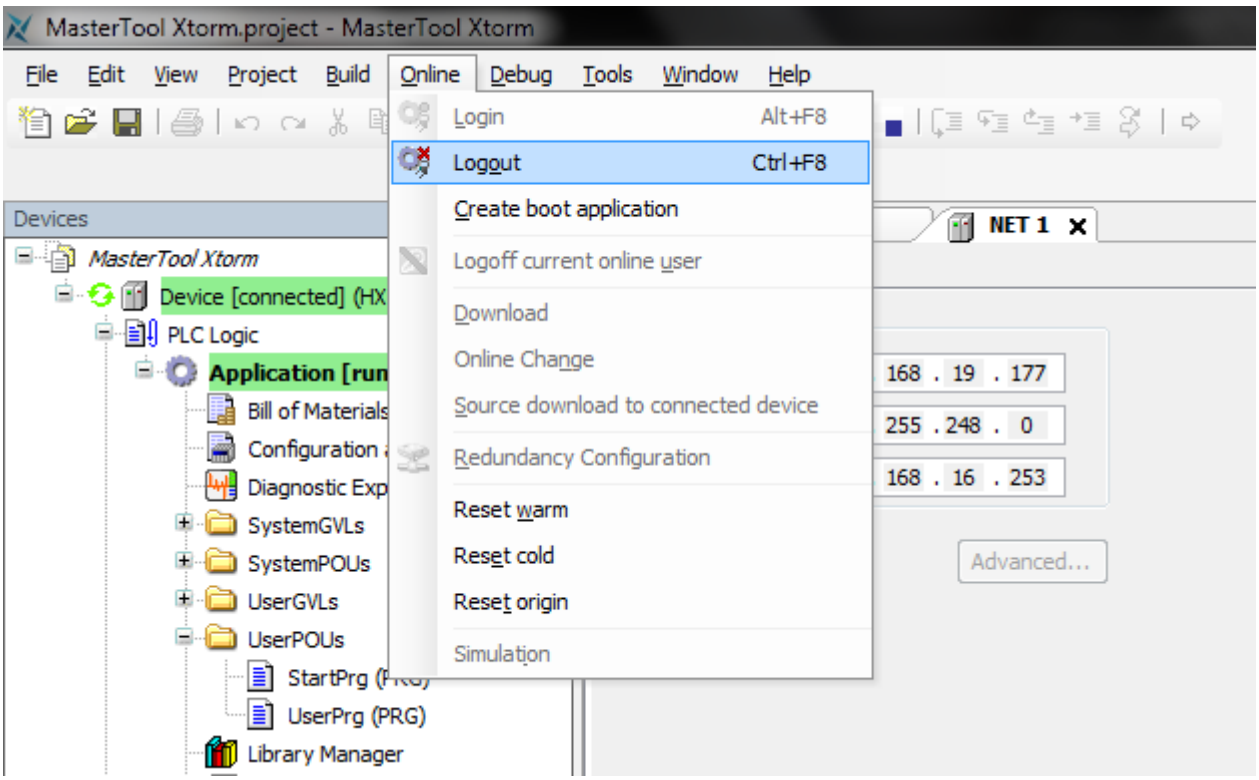


Figura 5-39. Interrompendo a Comunicação com a UCP

Modo Simulação

O MasterTool Xtorm possui um importante recurso de simulação que permite ao usuário testar sua aplicação sem a utilização do equipamento, conferindo maior agilidade no desenvolvimento do programa. No entanto, alguns recursos específicos, que dependem do hardware da UCP Hadron Xtorm, não são possíveis de serem simulados.

Seguem abaixo, os recursos indisponíveis no modo de simulação:

- Relógio RTC
- Varredura do Barramento
- Módulos de E/S
- Portas Seriais
- Comunicação Ethernet
- Protocolos de comunicação MODBUS
- Protocolos de comunicação DNP3
- Protocolo de comunicação MMS
- Protocolo de comunicação GOOSE
- Operações em cartão de memória
- Diagnósticos em variáveis
- Diagnostics Explorer
- Fila e agrupamento de eventos
- Outras funções que acessem o hardware da UCP

Por esta razão o modo simulação deve ser utilizado para testar a lógica da aplicação no que não depender de funções de acesso ao hardware. Estes recursos devem ser testados com o hardware para garantir o funcionamento da aplicação neste sentido.

Para alterar o MasterTool Xtorm para Modo Simulação é necessário selecionar esta opção no Menu Comunicação conforme Figura 5-40. Após isso é exibido um aviso no barra inferior do MasterTool Xtorm que indica que a ferramenta está operando em Modo Simulação.

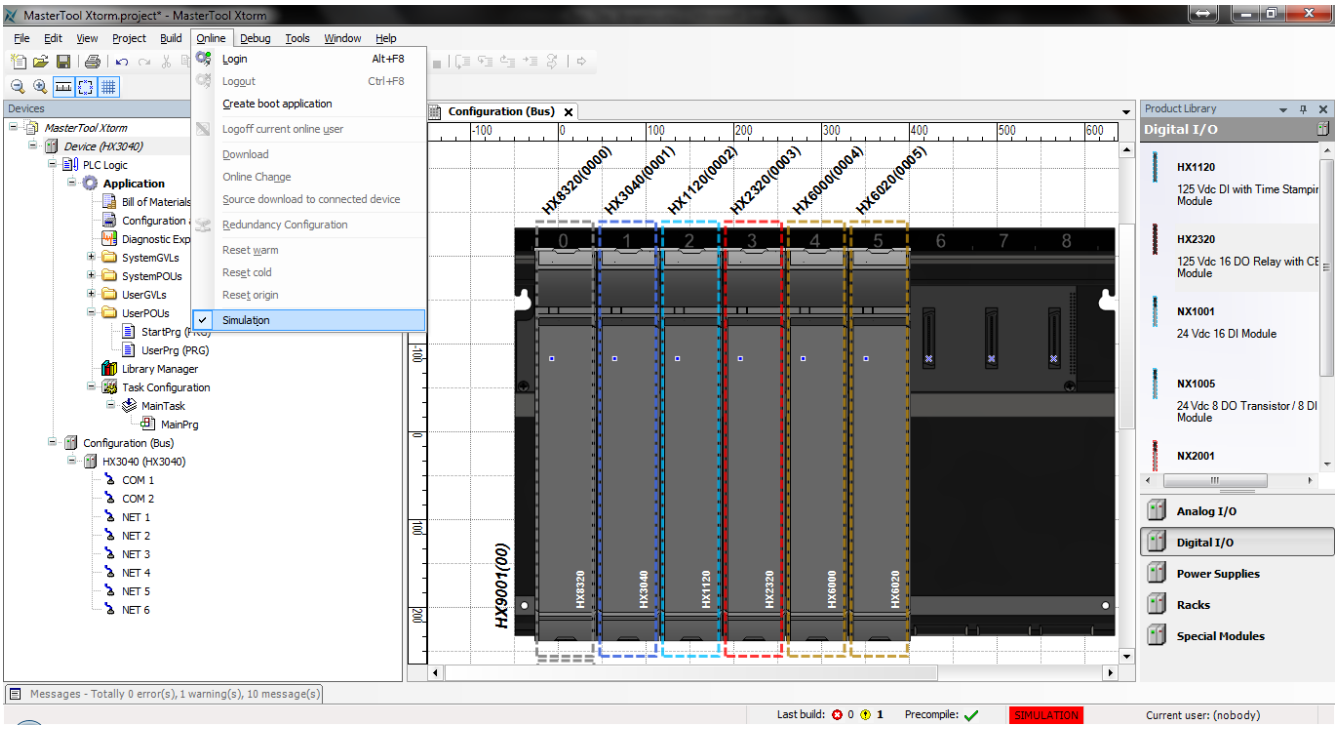


Figura 5-40. Modo Simulação

Em Modo Simulação a aplicação é executada em um dispositivo virtual no computador onde está instalado o MasterTool Xtorm. Por esta razão algumas características apresentadas estão relacionadas à arquitetura de hardware do computador e não da UCP da Série Hadron Xtorm. A principal característica neste sentido está relacionado ao formato dos dados nas áreas de memória de representação direta. O Modo Simulação trabalha com o formato little endian onde o primeiro endereço de memória é o menos significativo do dado. Por outro lado a UCP da Série Hadron Xtorm trabalha com o formato big endian onde o primeiro endereço de memória é o mais significativo do dado.

Neste caso um mesmo dado escrito por exemplo em %QD0, será escrito de forma diferente na simulação e na UCP da Série Hadron Xtorm. Se o dado escrito for 16#1234ABCD a distribuição dos dados na memória da UCP ficarão da seguinte maneira:

- %QW0 = 16#1234
- %QW2 = 16#ABCD
- %QB0 = 16#12
- %QB1 = 16#34
- %QB2 = 16#AB
- %QB3 = 16#CD

Para o mesmo dado escrito em %QD0 no Modo Simulação a distribuição dos dados na memória ficarão da seguinte maneira:

- %QW0 = 16#ABCD
- %QW2 = 16#1234
- %QB0 = 16#CD
- %QB1 = 16#AB
- %QB2 = 16#34
- %QB3 = 16#12

Tendo em vista estas diferenças e para facilitar o desenvolvimento da aplicação, utilizando os recursos do MasterTool Xtorm e da UCP da Série Hadron Xtorm é recomendado o uso de variáveis simbólicas. Neste caso as diferenças entre o Modo Simulação e o comportamento com a UCP da Série Hadron Xtorm não são verificadas. Portanto a melhor prática é evitar o uso de variáveis de representação direta sempre que possível para evitar o retrabalho quando desenvolver uma lógica que será testada em simulação e depois carregada em uma UCP.

O modo Simulação também pode ser utilizado para simular um projeto redundante, porém, terá as mesmas limitações citadas anteriormente, podendo ser testada apenas a lógica que não dependa do hardware. Neste caso, sempre serão executadas as POUs NonSkippedPrg e UserPrg, como se a UCP simulado fosse a UCP Ativa.

Upload do Projeto

A UCP da Série Hadron Xtorm possibilita a gravação de um projeto na memória do produto, o qual pode ser recuperado e reutilizado através do software MasterTool Xtorm.

Para armazenar um projeto na memória do produto, a UCP deve estar conectada (Login) e a opção de envio de projeto, juntamente com o aplicativo, deve estar selecionada.

Para recuperar o projeto previamente armazenado, deve ser selecionado as opções, conforme mostra a Figura 5-41.

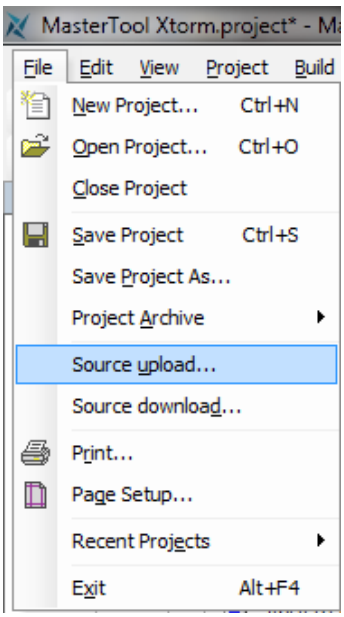


Figura 5-41. Opção de Upload de Projeto

Após, basta selecionar a UCP desejada e clicar em OK, conforme a Figura 5-42.

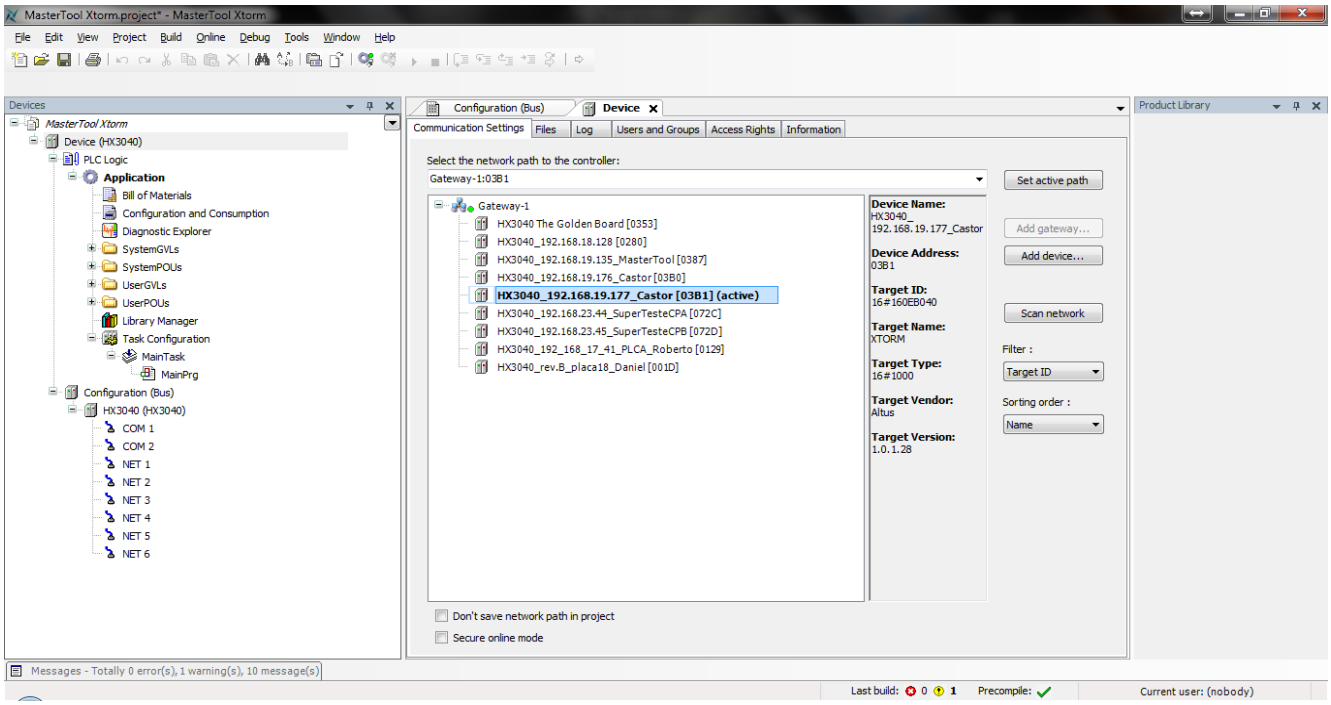


Figura 5-42. Seleciona a UCP

ATENÇÃO: O tamanho da área de memória para armazenar um projeto na UCP Hadron Xtorm está definido nas Características Gerais Específicas.

ATENÇÃO:
O Upload recupera o último projeto armazenado no controlador conforme descrito nos parágrafos anteriores. Caso ocorra apenas o carregamento para execução de um determinado aplicativo, o mesmo não poderá ser recuperado pelo procedimento de Upload.

Estados de Operação da UCP

Run

Quando uma UCP está em modo Run indica que todas as tarefas da aplicação estão em execução.

Stop

Quando uma UCP está em modo Stop, indica que as tarefas da aplicação estão paradas. O valor das variáveis nas tarefas são mantidas com o valor atual e variáveis de saída assumem valores definidos pelo usuário.

Quando uma UCP passa para modo Stop a partir do envio de uma aplicação, as variáveis nas tarefas da aplicação serão perdidas exceto as variáveis do tipo persistente. As variáveis de saída assumirão o valor definido pelo usuário e em seguida o valor das saídas passará para o estado seguro. Assim que a nova aplicação for carregada as variáveis de saída assumirão novamente o valor definido pelo usuário.

Breakpoint

Quando uma marca de depuração é atingida em uma tarefa, essa tarefa é interrompida. Todas as demais tarefas ativas na aplicação não irão ser interrompidas, elas continuarão a sua execução. Neste modo é possível percorrer um programa no modo Online. Um passo a passo pode ser executado e as posições das interrupções de depuração dependem do editor.

Notas:

Quando em modo de ponto de interrupção (BRKP), o serviço de barramento fica completamente parado, logo a UCP que controla o barramento terá alguns comportamentos específicos neste estado, são eles:

- A troca a quente de qualquer módulo não é suportado em Breakpoint, o módulo ficará sem parametrização caso ocorra uma troca quente nesse estado.
- Os diagnósticos de todos os módulos serão mantidos congelados na aplicação.
- O Botão de diagnóstico dos módulos não estará disponível, ou seja, caso o usuário aperte o botão no estado de breakpoint, nada acontecerá.

Exception

Quando uma UCP está em Exception indica que alguma operação indevida ocorreu em uma das tarefas ativas da aplicação. A tarefa que causou o Exception será suspensa e as demais tarefas irão para o modo Stop. Somente é possível tirar as tarefas desse estado e colocá-las em execução novamente após uma nova condição de partida da UCP. Portanto somente com um Reset a Quente, Reset a Frio, Reset Origem ou uma reinicialização da UCP coloca novamente a aplicação em modo Run.

Reset a Quente

Este comando coloca a UCP em modo Stop e inicializa todas as variáveis das tarefas da aplicação, exceto as variáveis dos tipos retentiva e persistente. As variáveis inicializadas com um valor específico assumirão exatamente este valor, as demais variáveis assumirão o valor padrão de inicialização (zero).

Reset a Frio

Este comando coloca a UCP em modo Stop e inicializa todas as variáveis das tarefas da aplicação, exceto as variáveis do tipo persistente. As variáveis inicializadas com um valor específico assumirão exatamente este valor, as demais variáveis assumirão o valor padrão de inicialização (zero).

Reset Origem

Este comando remove todas as variáveis das tarefas da aplicação, inclusive as variáveis do tipo persistente e apaga a aplicação da UCP.

Notas:

Reset: Quando um Reset é executado os breakpoints definidos na aplicação são desabilitados.

Comando: Para executar um comando de qualquer tipo de Reset é necessário estar em modo Online na UCP.

6. Redundância com UCP HX3040

Introdução

Este capítulo aborda a redundância de UCPs da Série Hadron Xtorm. Trata-se de uma redundância do tipo hot-standby, onde as UCPs são duplicadas no mesmo bastidor. Uma das UCPs normalmente está em estado Ativo e controlando o processo, enquanto a outra UCP normalmente está em estado Reserva, mantendo-se sincronizado com o controlador Ativo. Esta sincronização ocorre através de uma interface de sincronismo localizada no bastidor e dedicada para esta função. Esta interface é redundante, possuindo dois canais independentes (canal A e canal B). Caso ocorra uma falha na UCP Ativa, que o impeça de continuar controlando o processo, a UCP Reserva chaveia automaticamente para Ativa, em um tempo suficientemente baixo para não perturbar o processo, sem causar descontinuidades nas saídas que controlam o processo.

A redundância hot-standby é um método utilizado para aumentar a tolerância a falhas e, consequentemente, aumentar a disponibilidade do sistema de automação. A ideia básica é que nenhuma falha simples em componentes duplicados cause a interrupção do controle do processo.

A redundância hot-standby é muito aplicada em:

- Usinas de Geração de Energia
- Subestações

Cada UCP pode ter um ou mais protocolos de rede configurados para a comunicação com o centro de controle. Quando ocorre um erro com a UCP Ativa, a UCP reserva assume o controle das conexões ativas com o centro de controle. Com a duplicação destas redes, obtém-se uma disponibilidade ainda maior.

A redundância hot-standby de UCPs da Série Hadron Xtorm não prevê duplicação de módulos de E/S. Caso a redundância de módulos de E/S seja desejável, ela pode ser tratada em nível de aplicação, pelo usuário final. Por exemplo, o usuário pode duplicar ou até mesmo triplicar um módulo de entradas analógicas, e criar um algoritmo de votação para determinar qual das entradas irá ser considerada em determinado momento em sua aplicação.

A Figura 6-1 mostra um exemplo típico de arquitetura redundante com a UCP HX3040, onde:

1- Topologia de rede Ethernet

2- A configuração de cada UCP deve ser idêntica.

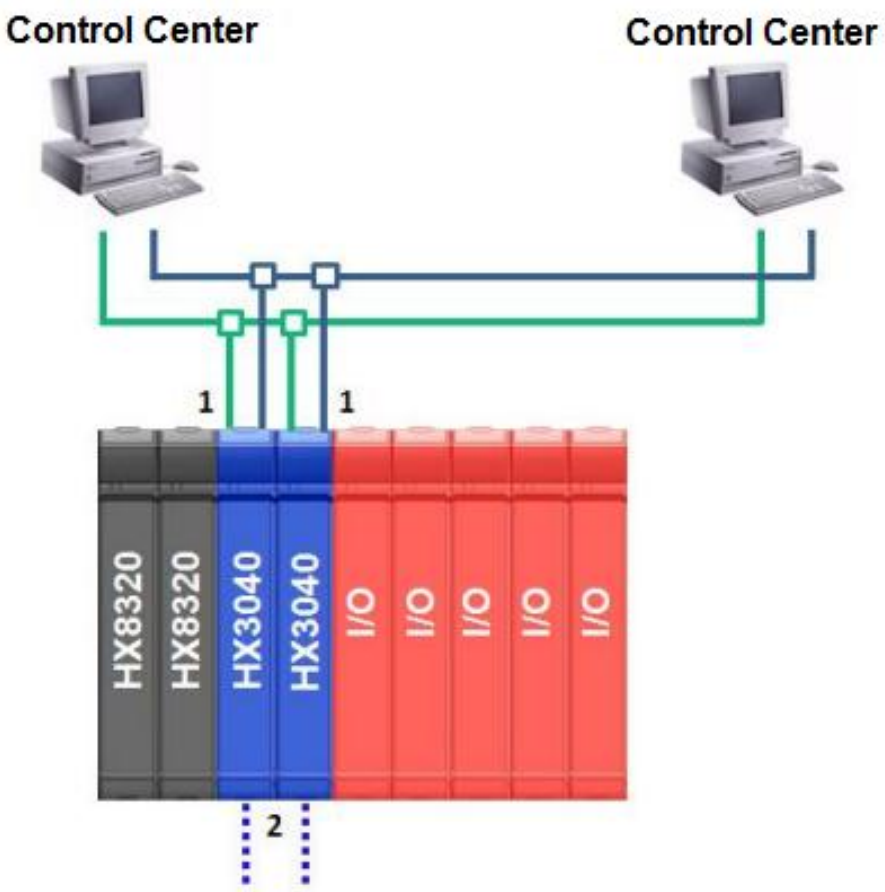


Figura 6-1. Exemplo de Arquitetura Redundante com UCP HX3040

Configurações para uma UCP Redundante

Para configurar uma UCP para o modo redundante, é necessário executar os seguintes passos:

- Criar um projeto selecionando a opção para uso de UCPs redundantes
- Para realizar esse procedimento, consultar o capítulo Wizard para Criação de um Novo Projeto Redundante
- Configurar as interfaces Ethernet. Para realizar esse procedimento, consultar o capítulo Configuração das Interfaces Ethernet
- Opcionalmente, instanciar protocolos DNP e MODBUS. As configurações de cada protocolo podem ser consultadas nas suas respectivas seções

As descrições de cada configuração estão relacionadas a seguir, neste capítulo.

Identificação de uma UCP HX3040

Ao longo deste manual, citaremos o par de UCPs redundantes como UCPA e UCPB. Identificamos uma UCP HX3040 que contém um projeto com redundância de UCP como:

- UCPA: a UCP inserida no posição 2 do barramento.
- UCPB: a UCP inserida no posição 3 do barramento.

Características Gerais

	Características Gerais de uma UCP redundante
UCPs permitidas	HX3040
Tipo de redundância	Hot-standby
Tolerância a falhas	Tolera, no mínimo, falha simples em equipamentos duplicados nos bastidores. Em casos específicos, pode tolerar múltiplas falhas.
5 estados de redundância	- Não-Configurado: estado inicial, também considerado quando a UCP está desligada ou não está executando a tarefa principal (MainTask). - Inicializando: estado temporário assumido depois de Não-Configurado, onde alguns testes determinarão o próximo estado (Inativa, Ativa, Reserva – ou de volta para Não-Configurado). - Inativa: estado atingido depois de determinados tipos de falhas ou para manutenção programada. - Ativa: controlando o processo do usuário. - Reserva: apto para chavear para Ativa e controlar o processo do usuário, caso haja uma demanda (ex: falha da UCP Ativa).
Principais falhas que causam switchover entre UCP Ativa e UCP Reserva. **UCP Reserva chaveia para Ativo e a UCP Ativa pode ir para Inativo ou Não-Configurado.**	- Falha de alimentação. - UCP (parada de execução da tarefa principal - MainTask). - Falha de ambos os canais de sincronismo entre as UCPs,
Comandos que causam switchover entre UCP Ativa e UCP Reserva	- Comandos via visor das UCPs . - Comandos recebidos do MasterTool Xtorm ou de um sistema SCADA, através desta UTR (local) ou da outra UTR (remota). - Comandos gerados pela aplicação do usuário, por exemplo, em função de outros diagnósticos (ex: falha de comunicação Ethernet), através desta UTR (local) ou da outra UTR (remota).
Principais falhas que impedem uma UCP de ficar no estado Reserva, ou assumir o estado Reserva. **Tais falhas fazem com que este UCP assuma os estados Não-Configurado ou Inativo.**	- Falha de alimentação. - UCP (parada de execução da tarefa principal - MainTask). - Falha em um dos canais de sincronismo (A ou B). - Falha no serviço de sincronismo de dados redundantes. - Projeto diferente daquele da UCP Ativa. - Versão de firmware incompatível com a UCP Ativa.
Comandos que tiram a UCP do estado Reserva	- Comandos via visor das UCPs redundantes. - Comandos recebidos do MasterTool Xtorm ou de um sistema SCADA, através desta UTR (local) ou da outra UTR (remota). - Comandos gerados pela aplicação do usuário, por exemplo, em função de outros diagnósticos (ex: falha de comunicação Ethernet), através desta UTR (local) ou da outra UTR (remota).
Tempo de switchover	- Tipicamente de um a três ciclos da MainTask, dependendo do estímulo para troca de estado (comando ou falha).
Switchover sem descontinuidades (bump-less)	- Um switchover não provoca descontinuidades nas saídas do controlador, nem em variáveis internas.
Overhead da redundância (consumo de UCP a cada ciclo da MainTask adicionado pela redundância)	- Valor máximo calculado automaticamente pelo MasterTool Xtorm e informado ao usuário, considerando uma lista de forçamentos redundantes vazia. - Valor médio típico de 15 ms para 100 Kbytes de dados redundantes e fila de 4500 eventos.
Visor da UCP	Entre outros diagnósticos, mostra o estado da redundância (Ativo, Reserva, Inativo, Não-Configurado, Inicializando). Através de opções no menu, permite comandos de switchover ou transições entre estados de redundância para manutenção.
Diagnósticos da redundância	- Indicam falhas tanto na UCPA como na UCPB, independente dos seus estados (Ativo ou Não-Ativo). - Evitam “falhas ocultas”. - Permitem manutenção rápida, o que é essencial para obter alta disponibilidade.
Comandos da redundância	- Permitem executar as mesmas ações do visor da UCP, além de outros comandos (ex: comandar switchovers). - Podem ser executados na UCP local, ou repassados para a outra UTR (remota) via canais de sincronismo entre as UCPs. - Podem ser recebidos a partir do MasterTool Xtorm ou de um sistema SCADA. - Podem ser executados a partir da aplicação do usuário.
Eventos da redundância	Registram modificações nos diagnósticos e comandos de redundância, com estampa de tempo, permitindo assim uma investigação das causas de um switchover.
IRIG-B, DNP3 e SNTP (Simple	Permite que os eventos tenham estampa de tempo precisa e ajustada à hora

Network Time Protocol)	mundial. Também sincroniza o relógio de tempo real da UCP para outras aplicações.
Telas no MasterTool para Diagnósticos, Comandos e Eventos do LOG de redundância	O programador MasterTool Xtorm disponibiliza telas especiais para comandos e diagnósticos da redundância, bem como para mostrar os LOGs de redundância.
Sincronização de dados de usuário	A cada ciclo da MainTask, UCPA e UCPB trocam diagnósticos e comandos. O usuário ainda possui 128 bytes de dados redundantes disponíveis dentro dos diagnósticos da UCP redundante, o sincronismo ocorre através dos canais de sincronização entre as UCPs. Desta maneira, uma UCP conhece os diagnósticos e comandos da outra.
Sincronização de variáveis redundantes	A cada ciclo da MainTask, a UCP Ativa copia dados redundantes para a UCP Não-Ativa através dos canais de sincronismo entre as UCPs. Dados não redundantes não são sincronizados.
Sincronização da lista de forçamentos redundantes	A cada ciclo da MainTask, a UCP Ativa copia a lista de forçamentos redundantes para a UCP Não-Ativa através dos canais de sincronismo entre as UCPs. Esta lista inclui apenas variáveis redundantes forçadas. Desta maneira, UCPA e UCPB poderão ter diferentes conjuntos de dados não redundantes forçados, pois estes forçamentos não são sincronizados. A fila de eventos DNP3 também é sincronizada neste procedimento da UCP, mesmo que a lista não esteja habilitada pelo usuário ela será sincronizada entre as UCPs mesmo não sendo usada pelo usuário. Para mais informações consultar capitulo Sincronização de Dados Redundantes.
Projeto único para UCPA e UCPB	Existe um projeto único comum para a UCPA e UCPB, gerado pelo MasterTool. O projeto é composto do projeto aplicativo (código executável) e do Project archive (código fonte).
Identificação de uma UCP	As UCPs são identificadas como UCPA ou UCPB automaticamente, dependendo da posição no qual estão inseridas. Ao inserir uma UCP na posição 2, ela recebe a designação de UCPA, e na posição 3, UCPB. Esta identificação não faz parte do projeto aplicativo gerado pelo MasterTool Xtorm. A identificação de cada UCP viabiliza a característica de ter um projeto único para UCPA e UCPB.
Sincronização automática do projeto	Se o projeto da UCP Ativa se tornar diferente do projeto da UCP Não-Ativa, o mesmo é copiado da UCP Ativa para a UCP Não-Ativa. Esta sincronização pode durar diversos ciclos da MainTask. Deve-se lembrar que o projeto é composto do projeto aplicativo (código executável) e do código fonte (Project archive), sendo ambos sincronizados.
Endereços IP privados para UCPA e UCPB	Pode-se conectar a uma UCP específica (UCPA ou UCPB) utilizando um endereço IP privado do mesmo, para, por exemplo, obter diagnósticos específicos dessa UCP, usando a interface Web ou a ferramenta de programação Mastertool. Para consulta a diagnósticos associados a protocolos de comunicação só será possível faze-lo na UCP ativa.
IP Ativo	Nome de uma estratégia que permite que clientes Ethernet se conectem a um servidor na UCP redundante, utilizando sempre o mesmo endereço de IP. Isto evita a necessidade de complexos scripts para mudar o endereço IP quando ocorrem switchovers devido à redundância. O Endereço de IP Ativo estará sempre associado à interface NET (i) da UCP Ativa.
NIC Teaming	Nome de uma estratégia que permite que duas interfaces Ethernet de uma UCP formem um par redundante compartilhando um mesmo endereço IP. Desta forma, podem-se construir redes Ethernet redundantes com maior facilidade, sem que os clientes conectados a um par NIC Teaming tenham de implementar complexos scripts para chavear endereços IP.
Tarefa de usuário cíclica	Somente uma tarefa é permitida. A tarefa é cíclica e tem o nome pré-definido: MainTask. A MainTask se destina a ser a tarefa principal, com a maior parte da lógica de usuário.
POUs principais	Na criação de um projeto redundante, o MasterTool gera automaticamente program POUs vazias que devem ser preenchidas pelo usuário. POUs associadas à MainTask: - NonSkippedPrg: esta program POU é executada nas duas UCPs (UCPA e UCPB), independente do estado de redundância (Ativo ou Não-Ativo). Destina-se ao gerenciamento de diagnósticos e comandos especiais. - UserPrg: esta program POU é executada somente na UCP Ativa, e destina-se ao controle do processo do usuário final. - StartPrg: esta program POU é executada nas duas UCPs (UCPA e UCPB), somente no primeiro ciclo da MainTask. POUs associadas à ProtTask: - MainPrg: POU que faz a chamada de outras assim como administra estas POUs. Não pode ser editada. - AlarmPrg: Esta POU estará associada aos alarmes utilizados na aplicação do usuário. Não pode ser editada. -EngineeringPrg: Assim como a AlarmPrg, é criada automaticamente e trata os processos de engenharia do projeto. Não pode ser editada.

Tabela 6-1. Características Gerais de uma UCP Redundante

Princípios de Funcionamento

Nesta seção, descrevem-se as funções de uma UCP redundante, seu comportamento e estados. Também são apresentados conceitos e restrições de programação e configuração que serão utilizados nos próximos capítulos.

Projeto Redundante Único

Fazendo uso da identificação descrita anteriormente, existe um único projeto para a UCP redundante, idêntico para os UCPA e UCPB.

Parâmetros de configuração que devem ser diferentes entre UCPA e UCPB (ex: endereços IP de interfaces Ethernet) aparecem duplicados no projeto da UCP redundante (um para UCPA, outro para UCPB). Cada UCP irá considerar aquele que lhe corresponde, após analisar sua identificação.

Estrutura do Projeto Redundante

Redundancy Template

O projeto de uma UCP redundante é criado automaticamente a partir de um modelo, denominado Redundancy Template. Além disso, são criadas tarefas e POUs básicas do tipo programa, conforme descrito nas próximas seções.

Tarefas Cíclicas: MainTask e ProtTask

O projeto de uma UCP redundante pode possuir duas tarefas: MainTask e ProtTask (criação opcional no Wizard), que são cíclicas. O usuário pode ajustar o intervalo de cada tarefa.

Programa MainPrg

A tarefa MainTask está associada a uma única POU do tipo programa, denominada MainPrg. O programa MainPrg é criado automaticamente.

O código do programa MainPrg é o seguinte, em linguagem ST:

```
IF isFirstCycle THEN
     StartPrg();
    isFirstCycle := FALSE;
ELSE
    NonSkippedPrg();
    IF fbRedundancyManagement.bActiveCPU THEN
          EngineeringPrg();
          AlarmPrg();
    UserPrg();
    END_IF;
END_IF;
```

A MainPrg chama outras POUs do tipo programa:

- No primeiro ciclo de execução, apenas StartPrg é chamada. Ela é executada tanto na UCP Ativa quanto na UCP Não-Ativa pois as variáveis da MainPrg não são redundantes
- Após o primeiro ciclo de execução, NonSkippedPrg sempre é chamada, pois é executada tanto na UCP Ativa quanto na UCP Não-Ativa. Já as POUs EngineeringPrg, AlarmPrg e UserPrg só são chamadas quando a condição " fbRedundancyManagement.bActiveCPU = TRUE" é verdadeira, ou seja, quando a UCP se encontra em estado Ativa

Portanto, o programa NonSkippedPrg será sempre executado em ambas as UCPs (UCPA e UCPB), independente do estado de redundância desta UCP. Já os programas EngineeringPrg, AlarmPrg e UserPrg serão executados somente na UCP que se encontra em estado Ativa.

Ao contrário do programa MainPrg, que não pode e não deve ser modificado, o usuário pode modificar os outros programas. Inicialmente, quando o projeto redundante é criado a partir do Redundancy Template, estes programas são criados “vazios”, mas o usuário poderá inserir código nos mesmos.

Programa StartPrg

Esta POU tem por objetivo ser executada somente uma única vez no primeiro ciclo de cada uma das UCPs.

Programa EngineeringPrg

Esta POU é bloqueada para edição e é gerada automaticamente pelo programa, ela mapeia as variáveis utilizadas para conversões de engenharia, característica da UCP de Conversão de Engenharia, descrito no capítulo Conversão de Engenharia.

Programa UserPrg

O principal objetivo deste programa, que é executado somente na UCP Ativa, é controlar o processo do usuário final.

Este programa normalmente atua sobre as variáveis redundantes, entre as quais encontram-se as variáveis de representação direta %I e %Q associadas ao sistema de E/S local. Para maiores informações, consultar o capítulo Programação de uma UCP Redundante - Configurações da MainTask.

Programa NonSkippedPrg

Este programa, que é executado em ambas UCPs (UCPA e UCPB) independente do estado de redundância, é tipicamente utilizado para funções como:

- Gerenciar condições de switchover normalmente não contempladas automaticamente pelas UCPs redundantes, que podem variar de usuário para usuário. Por exemplo, determinado usuário poderá executar um switchover para a UCP Reserva se a UCP Ativa não estiver se comunicando com o sistema SCADA, enquanto outro usuário pode não desejar um switchover nesta situação.
- Outras atividades que, por algum motivo, precisam ser executadas tanto na UCP Ativa como na UCP Reserva.

Para maiores informações, consultar o capítulo Programação de uma UCP Redundante - Configurações da MainTask - Programa NonSkippedPrg.

Programa AlarmPrg

Nesta POU é gerado o código automático responsável pela atribuição do alarme, alarme este que pode ser criado nas configurações da aba de Alarmes da UCP HX3040 .

Programa ProtPrg

A tarefa ProtTask está associada a uma única POU do tipo programa, denominada ProtPrg. O programa ProtPrg é criado somente caso o usuário escolha seu uso ao criar o projeto, para redundância.

O código do programa ProtPrg é o seguinte, em linguagem ST:

```
NonSkippedProtPrg();
IF fbRedundancyManagement.bActiveCPU THEN
   UserProtPrg();
END_IF
```

A ProtPrg chama outras POUs do tipo programa:

- A NonSkippedProtPrg sempre é chamada, pois é executada tanto na UCP Ativa quanto na UCP Não-Ativa e possui prioridade maior que a NonSkippedPrg. Já a POU UserProtPrg só é chamada quando a condição “ fbRedundancyManagement.bActiveCPU = TRUE” é verdadeira, ou seja, quando a UCP se encontra em estado Ativa.

Portanto, o programa NonSkippedProtPrg será sempre executado em ambas as UCPs (UCPA e UCPB), independente do estado de redundância desta UCP. Já o programa UserProtPrg será executado somente na UCP que se encontra em estado Ativa, esta POU tem prioridade maior em relação à UserPrg.

Ao contrário do programa ProtPrg, que não pode e não deve ser modificado, o usuário pode modificar os outros programas. Inicialmente, quando o projeto redundante é criado a partir do Redundancy Template, se selecionado sua criação, estes programas são criados “vazios”, mas o usuário poderá inserir código nos mesmos.

Programa UserProtPrg

O principal objetivo deste programa, que é executado somente na UCP Ativa, é controlar processos com prioridade superior a UserPrg do usuário final.

Esta POU é executada somente na UCP Ativa e destina-se ao controle do processo do usuário. A "UserProtPrg" é criada somente em projetos com redundância de UCP e possui uma prioridade maior em relação a "UserPrg". Para maiores informações, consultar o capítulo Programação de uma UCP Redundante Configurações da ProtTask.

Programa NonSkippedProtPrg

Este programa, que é executado em ambas UCPs (UCPA e UCPB) independente do estado de redundância com prioridade superior a NonSkippedPrg, é tipicamente utilizado para funções como:

- Atividades que, por algum motivo, precisam ser executadas tanto na UCP Ativa como na UCP Reserva com alta prioridade.

Para maiores informações, consultar o capítulo Programação de uma UCP Redundante - Programa NonSkippedProtPrg.

Variáveis Redundantes e Não-Redundantes

As variáveis de uma UCP redundante podem ser classificadas entre redundantes e não-redundantes. Variáveis redundantes são copiadas da UCP Ativa para a UCP Não-Ativa, no início de cada ciclo da MainTask, através dos canais de sincronismo entre as UCPs. Por outro lado, variáveis não-redundantes não são copiadas entre as UCPs e, portanto, podem ter valores diferentes na UCP Ativa e UCP Não-Ativa .

As variáveis não-redundantes são utilizadas para armazenar informações privativas de cada UCP (UCPA ou UCPB), tais como diagnósticos de drivers de comunicação e diagnósticos da UCP, incluindo os diagnósticos da redundância (estado da redundância desta UCP, etc.).

As variáveis redundantes dizem respeito às informações compartilhadas e relativas ao controle do processo. As variáveis associadas aos módulos de E/S são exemplos típicos de variáveis redundantes.

Variáveis %I Redundantes

Tipicamente as variáveis de entrada de representação direta %I são alocadas para armazenar os sinais lidos dos módulos de entrada digitais e analógicos do barramento local.

Nos projetos com redundância de UCP que possuem módulos de entrada no barramento local, temos uma faixa de variáveis %I que é automaticamente redundante. Esta faixa sempre inicia com a

variável %I de endereço 0 e se estende até a variável %I de maior endereço utilizado nos módulos de entrada locais.

Projetos com redundância de UCP que não possuem módulos de entrada no barramento local, não irão possuir variáveis de entrada de representação direta %I redundantes.

Variáveis %Q Redundantes

Tipicamente as variáveis de saída de representação direta %Q são alocadas para armazenar os sinais escritos nos módulos de saída digitais e analógicos do barramento local.

Nos projetos com redundância de UCP que possuem módulos de saída no barramento local, temos uma faixa de variáveis %Q que é automaticamente redundante. Esta faixa sempre inicia com a variável %Q de endereço 0 e se estende até a variável %Q de maior endereço utilizado nos módulos de saída locais.

Projetos com redundância de UCP que não possuem módulos de saída no barramento local, não irão possuir variáveis de saída de representação direta %Q redundantes.

Variáveis Simbólicas Redundantes e Não-Redundantes

Além das variáveis de representação direta (%I, %Q) que são alocadas automaticamente, o usuário pode declarar explicitamente variáveis simbólicas, dentro de POUs ou GVLs. O tamanho máximo permitido para alocação de variáveis simbólicas redundantes é de 512 Kbytes.

ATENÇÃO: Não se deve confundir variáveis simbólicas com variáveis simbólicas endereçadas através da diretiva AT. Variáveis simbólicas que utilizam a diretiva AT são nomes simbólicos atribuídos as variáveis de representação direta (%I, %Q e %M). Portanto, variáveis que utilizam a diretiva AT não alocam nenhuma memória de variáveis simbólicas.

Variáveis simbólicas são redundantes nos seguintes casos:

- Quando declaradas em POUs do tipo “programa” criadas na aplicação do usuário, com exceção dos programas NonSkippedPrg e NonSkippedProtPrg
- Quando declaradas em GVLs criadas na aplicação do usuário e estas GVLs marcadas como redundantes

Variáveis simbólicas são não-redundantes nos seguintes casos:

- Quando declaradas nos programas NonSkippedPrg e NonSkippedProtPrg, já descritos anteriormente
- Quando declaradas em POUs do tipo “função”. Observar que tais tipos de POUs normalmente deveriam alocar variáveis apenas na pilha (não estáticas), que consequentemente não precisam ser redundantes. Mesmo sabendo que o usuário pode declarar variáveis estáticas (VAR STATIC) dentro de POUs do tipo “função”, isto é considerado uma má prática de programação. Tais variáveis estáticas, caso sejam criadas, serão consideradas não redundantes
- Quando declaradas em POUs do tipo “bloco funcional”. Observar que a mera declaração de um “bloco funcional” não aloca memória (o que aloca memória é instanciar um Bloco Funcional)

Deve-se observar que instâncias de blocos funcionais, declaradas dentro de POUs do tipo programa ou dentro de GVLs, comportam-se como variáveis simbólicas, ou seja, alocam memória redundante. Da mesma maneira que variáveis simbólicas, quando instâncias de bloco funcional são declaradas nos programas NonSkippedPrg e NonSkippedProtPrg, ou ainda quando a GVL não é marcada como redundante, tais instâncias são não-redundantes.

Mapeamentos Múltiplos

Se o usuário desejar mapear as variáveis de comandos da redundância em mais de uma porta de comunicação (COMx ou NETx) será necessária a implementação de um controle pelo usuário na própria aplicação.

A lógica de controle a ser implementada deverá escrever nas variáveis de controle da redundância baseado nos valores das variáveis (comandos) provenientes de cada uma das portas de comunicação (COMx ou NETx). Além disso, a lógica de controle deve reinicializar as variáveis de comandos das portas de comunicação, uma vez que o controle da redundância só reinicializa suas próprias variáveis de comandos.

Segue um exemplo dessa implementação:

```
VAR
   var_comando_StandBy_relacao_Ethernet      : BOOL;
   var_comando_StandBy_relacao_Serial        : BOOL;
END_VAR

//Lógica para colocar a UCP Local para StandBy
IF var_comando_StandBy_relacao_Ethernet = TRUE THEN
   DG_HX3040_01.RedCmdLoc.bStandbyLocal:=TRUE;
   var_comando_StandBy_relacao_Ethernet:=FALSE;
END_IF
IF var_comando_StandBy_relacao_Serial = TRUE THEN
   DG_HX3040_01.RedCmdLoc.bStandbyLocal:=TRUE;
   var_comando_StandBy_relacao_Serial:=FALSE;
END_IF
```

Acima temos um exemplo de uma lógica em linguagem ST, onde o comando de switchover da redundância pode ser feito por duas variáveis, que provêm de portas de comunicação diferentes.

Onde:

var_comando_StandBy_relacao_Ethernet: variável do tipo Bool atribuída a um Coil da comunicação Ethernet que realizará o comando para colocar a UCP Local para StandBy.

var_comando_StandBy_relacao_Serial: variável do tipo Bool atribuída a um Coil da comunicação Serial que realizará o comando para colocar a UCP Local para StandBy.

DG_HX3040_01.tRedundancy.RedCmdLoc.bStandbyLocal: este comando produz uma ação equivalente ao comando de “Trocar para Reserva” disparado através do menu do visor, na UCP local.

Estruturas de Dados de Diagnósticos, Comandos e Usuário

Cada UCP dispõe de diversas estruturas de dados relacionadas com a redundância. As seguintes estruturas pertencem à variável simbólica DG_HX3040_01, alocada automaticamente e disponível para o usuário:

- RedDgnLoc: contém diagnósticos da UCP local relacionados com a redundância, como por exemplo, o estado da redundância desta UCP
- RedDgnRem: é uma cópia de RedDgnLoc da outra UCP, recebida via canais de sincronismo da redundância. Desta forma, esta UCP (local) tem acesso aos diagnósticos da outra UCP (remota)
- RedCmdLoc: contém comandos que devem ser aplicados nesta UCP (quando têm sufixo Local) ou na outra UCP (quando têm sufixo Remote). Por exemplo o campo StandbyLocal desta estrutura de dados corresponde a um comando que deve ser executado nesta UCP (local), enquanto o campo StandbyRemote corresponde a um comando que deve ser executado na outra UCP (remota)
- RedCmdRem: é uma cópia de RedCmdLoc da outra UCP, recebida via canais de sincronismo da redundância, utilizada somente para visualização ou informação
- RedUsrLoc: contém 128 bytes de dados preenchidos livremente pelo usuário (ex: diagnóstico de comunicação com um sistema SCADA). Estes 128 bytes de dados serão intercambiados com a outra UCP
- RedUsrRem: é uma cópia de RedUsrLoc da outra UCP, recebida via canais de sincronismo

Na seção Manutenção, as seguintes subseções fornecerão maiores detalhes sobre as Estrutura de Diagnósticos da Redundância:

- Diagnósticos da Redundância

- Comandos da Redundância
- Informações do Usuário Trocados entre UCPA e UCPB

Serviços de Sincronização Cíclicos através de Canais de Sincronismo de Redundância

Esta seção descreve os serviços de sincronização que ocorrem ciclicamente em uma UCP redundante, entre UCPA e UCPB, através de dois canais de sincronismo interno feito através do barramento entre as UCPs. Os canais de sincronismo tem por função sincronizar variáveis redundantes, diagnósticos, área de memoria de usuário redundante, fila de eventos, sincronização de projetos e comandos.

Estes serviços são executados no início de cada ciclo da MainTask, e na sequência em que aparecem abaixo, ou seja:

- Primeiro, executa-se o serviço Troca de Diagnósticos e Comandos
- Segundo, executa-se o serviço Sincronização de Dados Redundantes

Estes serviços são executados tanto em STOP quanto em RUN. Modificações e forçamentos feitos em variáveis através da monitoração no Mastertool são sincronizadas entre as UCPs mesmo em STOP.

Troca de Diagnósticos e Comandos

Este serviço é responsável pelo intercâmbio das seguintes estruturas de dados, em cada ciclo da MainTask:

- Copiar RedDgnLoc da UCPA para RedDgnRem da UCPB
- Copiar RedCmdLoc da UCPA para RedCmdRem da UCPB
- Copiar RedUsrLoc da UCPA para RedUsrRem da UCPB
- Copiar RedDgnLoc da UCPB para RedDgnRem da UCPA
- Copiar RedCmdLoc da UCPB para RedCmdRem da UCPA
- Copiar RedUsrLoc da UCPB para RedUsrRem da UCPA

Este serviço sempre será executado, desde que hajam duas UCPs no barramento (nos slots corretos) e desde que a aplicação presente nas duas UCPs seja redundante.

Sincronização de Dados Redundantes

Este serviço é responsável pela transferência de dados da UCP Ativa para a UCP NÃO-Ativa, o que inclui:

- Fila de eventos
- Variáveis redundantes (simbólicas e de representação direta - %I e %Q)
- Lista de forçamentos de variáveis redundantes

Para que este serviço seja executado, diversas condições devem ser satisfeitas:

- O serviço de sincronização anterior deste ciclo da MainTask (Troca de Diagnósticos e Comandos) deve ter sido completado com sucesso
- Caso esta UCP esteja em estado Ativo, a outra deve estar em estado Não-Ativo. Por outro lado, caso esta UCP esteja em estado Não-Ativa, a outra deve estar em estado Ativa
- Os projetos das duas UCPs devem estar idênticos. Caso uma UCP seja trocada (troca-a-quente) e a nova UCP não contenha um programa idêntico, essa sincronização somente será executada depois que o Sincronismo de Projeto substituir o programa da nova UCP

ATENÇÃO:
A sincronização da fila de eventos é realizada sempre para todos os registros de eventos contidos nela, independentemente se estes eventos foram gerados a partir de variáveis redundantes ou de variáveis não redundantes.

ATENÇÃO:
A lista de forçamentos redundantes contém apenas informações de forçamentos sobre variáveis redundantes. Em cada uma das UCPs (UCPA e UCPB), pode existir uma lista diferente de forçamentos sobre variáveis não redundantes. Forçamentos de variáveis não redundantes não são sincronizados entre os UCPs. Para forçar uma variável não redundante, deve-se fazer o Login no UCP desejado e forçá-la.

Serviços de Sincronização Esporádicos Através dos Canais de Sincronismo de Redundância

Os seguintes serviços de sincronização são executados de forma esporádica. Ou seja, não são executados a cada ciclo da tarefa MainTask, mas outra tarefa do sistema executa estes serviços esporádicos em segundo plano.

Sincronização de Projetos

Este serviço é responsável por manter sincronizados os projetos das UCPs Ativa e Não-Ativa. Isto ocorre apenas quando os projetos estão diferentes .

A sincronização é sempre no sentido da UCP Ativa para a UCP Não-Ativa e basta que um dos canais de sincronismo esteja operacional para que este serviço seja executado.

Quando a sincronização estiver habilitada, serão sincronizados os seguintes arquivos e serviços:

- Projeto aplicativo (código executável)
- Project archive (código fonte)
- Usuários e grupos
- Direitos de acesso
- Trace

O serviço de sincronização irá iniciar em até trinta segundos, após uma das UCPs entrar no estado Ativo, e, após seu início, irá checar o CRC dos projetos a cada cinco segundos.

Quando uma sincronização for iniciada, a UCP Não-Ativa, se estiver em RUN, irá para o modo STOP, no estado de Não Configurado. Após a transferência de todos os arquivos necessários, a UCP Não-Ativa irá para o mesmo modo da UCP Ativa (STOP ou RUN), no estado de Inicializando.

O tempo que a sincronização levará para ser efetuada dependerá do tamanho do projeto. Em média, a taxa de transferência entre os canais de sincronismo é de aproximadamente 500 Kbytes/s.

Caso a sincronização seja interrompida por falha de comunicação entre os canais de sincronismo durante a transferência dos arquivos da UCP Ativa para o Não-Ativo, o procedimento será abortado e reinicializado quando a comunicação for restaurada. Somente após a conclusão de todo o procedimento a UCP Não-Ativa irá para o mesmo modo que a UCP Ativa (STOP ou RUN).

Além de manter sincronizados os projetos, a Sincronização do Projeto também irá impedir que a UCP Não-Ativa assuma estados posteriores a Inicializando, caso o CRC esteja diferente ou algum Online Change esteja pendente na UCP Ativa.

ATENÇÃO:
Uma sincronização de projeto terá os mesmos efeitos de um download na UCP Não-Ativa.

Redes Ethernet Redundantes com NIC Teaming

Cada UCP pode ter um ou mais protocolos de rede configurados para a comunicação com o centro de controle.

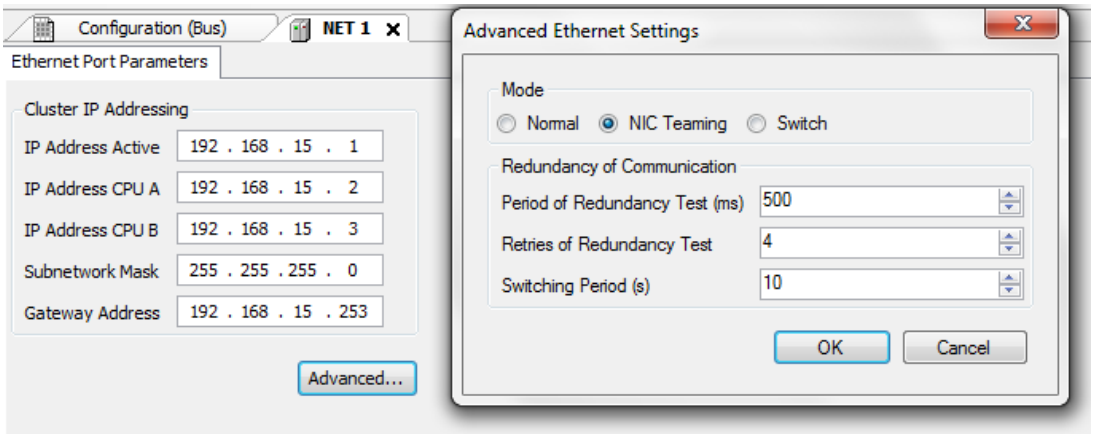


Figura 6-2. Exemplo de rede Ethernet redundante, com NIC Teaming

Neste exemplo a UCP HX3040 conecta-se à rede de supervisão (SCADAs). As duas portas Ethernet da UCP HX3040 (NET 1 e NET 2) formam um par redundante NIC Teaming, interligadas em dois switches diferentes (Switch 1 e Switch 2). Em algum ponto, estes dois switches devem ser interligados, para que exista conexão entre as duas portas NIC Teaming e disponibilidade ainda maior (contra falhas duplas).

Tal arquitetura Ethernet possibilita elevada disponibilidade de comunicação do sistema, sendo fortemente indicada para transpor falhas nas portas Ethernet, em cabos e em switches.

ATENÇÃO: Quando duas interfaces Ethernet formam um par redundante NIC Teaming, a configuração dos parâmetros básicos e adição de protocolos de comunicação (MODBUS ,DNP3, etc.) possível é feita apenas na primeira interface, sendo que os parâmetros da segunda interface ficam bloqueados para edição.

Maiores detalhes sobre configuração e diagnósticos de portas NIC Teaming são fornecidos na seção Configurações das Portas Ethernet da UCP HX3040.

Métodos de Troca de IP

Um projeto com UCP redundante da Série Hadron Xtorm dispõe de um método para a troca de IP das portas NET 1 a NET 6. Esse método, chamado de IP Ativo, define o comportamento das portas, no que se refere ao IP da mesma, de acordo com o estado atual de redundância da UCP (Ativo ou Não-Ativo) e a identificação da UCP em questão (UCPA ou UCPB).

Será necessário configurar três endereços IP por interface NET. Caso a NET esteja trabalhando em NIC Teaming ou modo Switch, será necessário configurar os três endereços IPs para o par de NETs ao invés de fazer uma configuração individual.

IP Ativo

Esse método é o utilizado nas NETs da UCP HX3040 redundante. Nesse método há um IP para a UCP Ativa e mais dois IPs, um para a UCPA e outro para a UCPB. Nas NETs da UCP HX3040 redundante, o Endereço de IP Ativo será associado à interface da UCP Ativa.

O protocolo de comunicação MODBUS, tanto cliente quanto servidor, utilizam o IP local da UCP, já para o protocolo DNP3 o IP Ativo é usado para se comunicar.

Já para a comunicação com o Mastertool, deve-se usar o endereço de IP específico da UCPA ou UCPB.

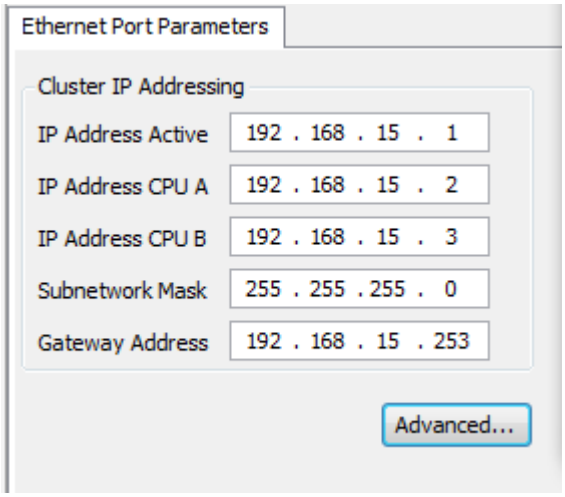


Figura 6-3. Método IP Ativo – HX3040 Redundante

Parâmetros que devem ser configurados no método de IP Ativo para as NETs de uma UCP HX3040 Redundante:

- IP Address Active: endereço IP associado à interface quando a UCP estiver em estado Ativo
- IP Address CPU A: endereço para comunicação com a UCPA, independente do seu estado atual
- IP Address CPU B: endereço para comunicação com a UCPB, independente do seu estado atual
- Subnetwork Mask: máscara de subrede das UCPs no barramento Ethernet
- Gateway Address: endereço do gateway da subrede do barramento Ethernet

Uso Combinado de NIC Teaming e IP Ativo

Caso determinado par de portas forme um par NIC Teaming em uma UCP redundante, estas portas podem implementar, ao mesmo tempo, as estratégias NIC Teaming e IP Ativo.

Por exemplo, se as portas NET 1 e NET 2 da UCP HX3040 formam um par NIC Teaming, então:

- IP Address CPU A: endereço IP das portas NET 1 + NET 2 da UCP HX3040 que se encontra na posição 2 do barramento
- IP Address CPU B: endereço IP das portas NET 1 + NET 2 da UCP HX3040 que se encontra na posição 3 do barramento
- IP Address Active: endereço IP das portas NET 1 + NET 2 daquela UCP HX3040 que se encontra no estado Ativo

Desta maneira, associa-se a excelente disponibilidade da estratégia NIC Teaming com a praticidade da estratégia de IP Ativo, que dispensa scripts em sistemas SCADA ou em outros clientes conectados a servidores na UCP Ativa.

Estados de uma UCP Redundante

Em um sistema redundante, uma UCP (UCPA ou UCPB) pode assumir os seguintes estados:

- Ativo
- Reserva
- Inativo
- Não-Configurado
- Inicializando

ATENÇÃO:
Frequentemente este manual utilizará a designação “Não-Ativo” para qualquer estado diferente de Ativo, ou seja, para designar qualquer um dos outros quatro estados: Reserva, Inativo, Não-Configurado e Inicializando.

A seguir, estes cinco estados são descritos brevemente. Maiores detalhes sobre os estados de uma UCP redundante serão abordados na seção Transições entre Estados de Redundância, ao descrever a máquina de estados e as causas das transições entre os mesmos.

Estado Não-Configurado

Este é o estado de redundância inicial. A UCP encontra-se neste estado de redundância:

- Por convenção, enquanto a UCP estiver desligada
- Antes de iniciar a MainTask
- Antes de chavear para o estado Inicializando
- Caso ocorra uma reinicialização através de um comando como Reset a Quente, Reset a Frio ou Reset Origem

Caso a MainTask esteja executando no estado Não-Configurado, as seguintes tarefas são executadas:

- Os módulos no barramento não são controlados pela UCP
- Os serviços de sincronização cíclicos são executados (ver seção Serviços de Sincronização Cíclicos através de Canais de Sincronismo de Redundância desde que as condições para sua execução estejam presentes
- Os serviços de sincronização esporádicos também podem ser executados (ver seção Serviços de Sincronização Esporádicos Através dos Canais de Sincronismo de Redundância)

A UCP ficará bloqueada no estado Não-Configurado se o outra UCP estiver em estado Ativo e o projeto deste UCP for diferente do projeto da UCP Ativa. Caso esta situação não se confirme, ocorre uma transição do estado Não-Configurado para o estado Inicializando assim que chegar uma requisição de configuração.

Algumas vezes, a UCP entra no estado Não-Configurado já tendo recebido uma requisição de configuração automática, não sendo necessária uma nova requisição para mudar para o estado Inicializando. Isto ocorre, por exemplo, na energização da UCP.

Outras vezes, o usuário deverá requisitar manualmente esta configuração, por exemplo, usando o menu do visor da UCP. Requisições de configuração manuais tipicamente são necessárias quando alguma manutenção por parte do usuário é necessária, antes de sair do estado Não-Configurado, por exemplo, se a UCP atingiu o estado Não-Configurado devido a alguma falha.

Depois de sair do estado Não-Configurado, a UCP pode voltar a este estado, devido a eventos tais como:

- Reinicialização (Reset a Quente, Reset a Frio ou Reset Origem)
- Desligamento da UCP
- Diferença de projeto entre esta UCP e a UCP Ativa

Estado Inicializando

Diferente de todos os outros quatro estados que podem ter duração indeterminada, o estado Inicializando é temporário, durando poucos segundos. Este estado é sempre alcançado a partir do estado Não-Configurado, através de uma requisição de configuração.

Ao entrar no estado Inicializando, diversas ações, testes e verificações são executados, para decidir qual será o próximo estado:

- A UCP apenas escuta o barramento local, sem interferir no controle dos módulos
- Verifica se a identificação da UCP está correta (deve ser UCPA ou UCPB, de acordo com o onde a UCP está inserido)

- Verifica se há problemas nos parâmetros de configuração extraídos do projeto MasterTool
- Os serviços de sincronização cíclicos são executados (ver seção Serviços de Sincronização Cíclicos através de Canais de Sincronismo de Redundância), desde que as condições para sua execução estejam presentes
- Verifica compatibilidade de versões de firmware entre os dois UCPs
- Verifica igualdade de projetos entre os dois UCPs
- Caso a outra UCP esteja em estado Ativo, verifica a possibilidade de estabelecer comunicação passiva com o mesmo, usando o barramento local dos módulos. O modo passivo serve para testar os circuitos de transmissão e recepção e o meio físico, para evitar a ocorrência de uma falha oculta
- Caso a outra UCP esteja em estado desconhecido devido a falhas no canal de sincronismo de redundância, verifica a possibilidade de estabelecer comunicação passiva com o mesmo, usando o barramento local dos módulos

Dependendo do resultado destas verificações e testes, a UCP pode ir do estado Inicializando para qualquer um dos outros quatro estados.

Estado Ativo

Neste estado, a UCP controla o processo automatizado, usando o programa UserPrg, executado somente neste estado. A UCP Ativa também controla os módulos de E/S, mantendo-os operacionais, lendo as entradas e escrevendo as saídas.

A UCP Ativa também verifica seus diagnósticos internos e requisições de switchover do usuário para determinar se um switchover é necessário. A UCP normalmente só sairá do estado Ativo se souber que a outra UCP está em estado Reserva, e apta para assumir como Ativo.

No entanto, existem algumas situações em que a UCP Ativa poderá sair do estado Ativo mesmo sem ter certeza de que a outra UCP está no estado Reserva (por exemplo, se esta UCP for removida do bastidor).

Estado Reserva

Neste estado, a UCP está pronta para chavear para o estado Ativo, caso haja uma demanda para isso, tal como uma falha na UCP Ativa atual.

A UCP Reserva também verifica seus próprios diagnósticos, e poderá chavear para o estado Não-Configurado ou Inativo em função de algumas falhas.

O controle do barramento é mantido em estado passivo. O modo passivo serve para testar os circuitos de transmissão e recepção e o meio físico, para evitar a ocorrência de uma falha oculta. Falhas totais causam um chaveamento para o estado Inativo.

Estado Inativo

Este estado normalmente é alcançado depois de alguns tipos de falhas, ou devido a alguma solicitação manual antes de uma manutenção programada.

O controle do barramento é mantido em estado passivo. O modo passivo serve para testar os circuitos de transmissão e recepção e o meio físico, para evitar a ocorrência de uma falha oculta.

Antes de deixar este estado, primeiro deve-se corrigir as falhas diagnosticadas ou executar as manutenções programadas, que causaram a transição para o estado Inativo. Depois, deve-se causar uma transição para o estado Não-Configurado já solicitando uma configuração, para logo em seguida chavear para o estado Inicializando. Depois do estado Inicializando, a UCP pode:

- Retornar ao estado Inativo se determinados tipos de falhas persistem
- Retornar ao estado Não-Configurado para outros tipos de falhas
- Ir para o estado Reserva se o outra UCP está no estado Ativo
- Ir para o estado Ativo se o outra UCP não está no estado Ativo

Comandos do menu de redundância do visor da UCP

Esta seção descreve as funções dos comandos do menu de redundância, acessado pelo visor da UCP.

O comando “Trocar para Reserva” tem as seguintes funções:

- Solicitar um chaveamento do estado Ativo para o estado Reserva, o que pode ser útil para executar alguma manutenção programada na UCP Ativa. Depois que o UCP Ativa chaveia para Reserva (e consequentemente a UCP Reserva chaveia para Ativo), é possível chaveá-lo de Reserva para Inativo usando o comando “Trocar para Inativo”, e então executar a manutenção programada no estado Inativo
- Solicitar uma configuração que provoca um chaveamento do estado Não-Configurado para o estado Inicializando, tipicamente depois de reparar falhas que provocaram a transição para o estado Não-Configurado. Depois do estado Inicializando, normalmente espera-se que a UCP vá para o estado Reserva (ou Ativo, se o outra UCP não estiver no estado Ativo)
- Solicitar um chaveamento do estado Inativo para o estado Não-Configurado já solicitando uma configuração. Isto ocorre tipicamente depois de corrigir falhas que provocaram a transição para o estado Inativo. Depois do estado Não-Configurado, a configuração já deve levar ao estado Inicializando. Depois do estado Inicializando, normalmente espera-se que a UCP vá para o estado Reserva (ou Ativo, se o outra UCP não estiver no estado Ativo)

O comando “Trocar para Inativo” solicita um chaveamento do estado Reserva para o estado Inativo, o que pode ser útil para executar alguma manutenção programada na UCP Reserva. Depois desta manutenção, pode-se utilizar o comando ”Trocar para Reserva” para tentar voltar ao estado Reserva, passando pelos estados Não-Configurado e Inicializando (ver descrição anterior das funções do comando ”Trocar para Reserva”).

ATENÇÃO: Existem maneiras alternativas de gerar os mesmos efeitos dos comandos ”Trocar para Reserva” e “Trocar para Inativo”. Podem-se utilizar comandos gerados por esta UCP ou pela UCP remota, conforme descrito preliminarmente na seção Estruturas de Dados de Diagnósticos, Comandos e Usuário. Uma descrição mais detalhada destes comandos pode ser encontrada na seção Comandos da Redundância.

Transições entre Estados de Redundância

A figura seguinte mostra a máquina de estados da redundância, ilustrando todas as possíveis transições entre estados de redundância.

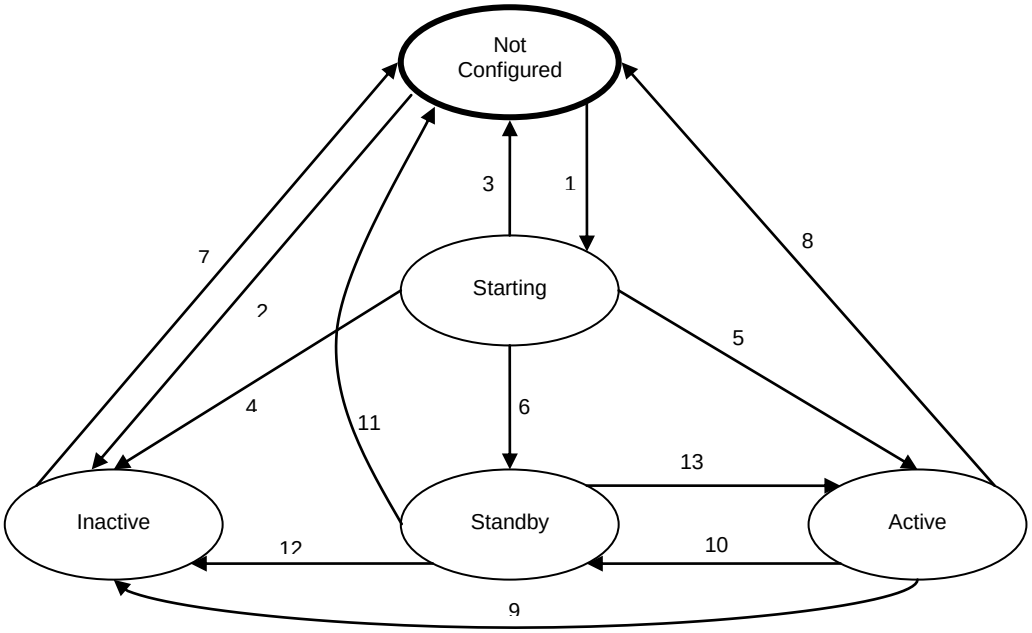


Figura 6-4. Máquina de Estados da Redundância

As seguintes subseções descreverão todas estas transições, e as causas que podem dispará-las. Para interpretar corretamente o funcionamento desta máquina de estados, é preciso estabelecer algumas regras e sequências:

- Transições que se originam do mesmo estado devem ser avaliadas na sequência dada pelo número da transição. Por exemplo, as transições 3, 4, 5 e 6 se originam do estado Inicializando. Neste exemplo, avalia-se primeiro a transição 3, depois a 4, depois a 5 e finalmente a 6. Caso a transição 3 seja disparada, as transições 4, 5 e 6 nem serão avaliadas
- Dentro de uma subseção específica que descreve uma transição, diversas condições podem disparar esta transição. Estas condições devem ser avaliadas na ordem que aparecem na subseção. Qualquer uma destas condições que se tornar verdadeira pode causar a transição. Se uma condição causar a transição, as próximas condições não serão avaliadas
- As transições ocorrem tanto com a MainTask em STOP como em RUN
- Em diversos casos, mencionam-se transições causadas pelos comandos disponíveis no menu do visor da UCP. Deve-se lembrar de que existem alternativas para estes comandos, que são os comandos internos provenientes desta UCP ou da outra UCP (através do link interno de redundância). Tais comandos estão melhores detalhados na seção mencionada preliminarmente Estruturas de Dados de Diagnósticos, Comandos e Usuário. Nas subseções seguintes, por simplicidade, estes comandos alternativos não serão citados, mas deve-se ter consciência que eles podem causar as mesmas transições do comando disponível no menu do visor
- Ao executar Reset Origem a aplicação é removida, e sendo assim, o estado da redundância é removido do visor

Transição 1 - Não-Configurado para Inicializando

- Um pedido de configuração já existia ao entrar no estado Não-Configurado. Isto ocorre na energização da UCP, e também em outras situações, descritas nas próximas subseções
- O comando de STAND-BY foi solicitado durante o estado Não-Configurado. Isto causa um pedido de configuração manual. O usuário tipicamente solicita o comando de STAND-BY depois de reparar falhas que anteriormente levaram esta UCP ao estado Não-Configurado ou Inativo

Transição 2 – Não-Configurado para Inativo

- Ocorreu alguma exceção de software nesta UCP (cão-de-guarda, access violation, illegal instruction)

Transição 3 - Inicializando para Não-Configurado

- Esta UCP foi desligada ou reinicializada (Reset a Quente, Reset a Frio ou Reset Origem).
- A UCP está inserida numa posição incorreta
- Existem erros lógicos de configuração no projeto recebido do MasterTool Xtorm
- A outra UCP está no estado Ativo, e a versão de firmware desta UCP é incompatível com a versão de firmware da UCP Ativa
- A outra UCP está em estado Ativo, e o projeto desta UCP é diferente do projeto da UCP Ativa. Além de ir para o estado Não-Configurado, uma solicitação de configuração é feita. Desta maneira, depois que os projetos forem sincronizados, a UCP sai automaticamente do estado Não-Configurado para o estado Inicializando

Transição 4 - Inicializando para Inativo

- Ocorreu alguma exceção de software nesta UCP (cão-de-guarda, access violation, illegal instruction)
- Algum dos canais de sincronismo do link interno de redundância (canal A ou B) está em falha, e esta UCP sabe que esta falha foi causada por componentes de hardware ou software internos (falhas internas do canal A ou B)
- A outra UCP está em estado Ativo. No entanto, não é possível sincronizar os dados redundantes ou a lista de forçamentos redundantes
- O estado da outra UCP não pode ser descoberto via link interno de redundância, mas esta UCP consegue monitorar atividade no barramento. Desta maneira, parece que a outra UCP está controlando o processo, embora o link interno de redundância não esteja funcionando adequadamente para ter certeza disso
- O estado RUN/STOP da aplicação está diferente da UCP Ativa. Por alguma razão de falha, a UCP não sincronizou o estado RUN/STOP da aplicação com a outra UCP Ativa

Transição 5 - Inicializando para Ativo

- A outra UCP encontra-se em estado Não-Ativo. Antes de fazer esta transição, esta condição deve permanecer verdadeira durante algum tempo. Quando a UCPA e UCPB são energizadas simultaneamente, a UCP que terminar a inicialização do sistema antes assume como Ativo
- O estado da outra UCP não pode ser descoberto via link interno de redundância, e além disso esta UCP não consegue monitorar atividade no barramento. Portanto, parece realmente que a outra UCP está ausente ou fora de execução (entrou em exceção de software). Por razões de segurança, a UCP em reserva irá chavear para Ativo. Esta condição deve se manter durante algum tempo antes de fazer esta transição

Transição 6 - Inicializando para Reserva

- A outra UCP encontra-se em estado Ativo e os serviços de sincronização de dados redundantes (fila de eventos, variáveis redundantes e lista de forçamentos de variáveis redundantes) estão funcionando corretamente

Transição 7 - Inativo para Não-Configurado

- Esta UCP foi desligada ou reinicializada (Reset a Quente, Reset a Frio ou Reset Origem)
- O comando requisitando transição para STAND-BY foi executado. Além de ir para o estado Não-Configurado, uma solicitação de configuração é feita. Desta maneira, a UCP sai automaticamente do estado Não-Configurado para o estado Inicializando. O usuário tipicamente solicita este comando depois de reparar uma falha que levou a UCP anteriormente ao estado Inativo

Transição 8 - Ativo para Não-Configurado

- Esta UCP foi desligada ou reinicializada (Reset a Quente, Reset a Frio ou Reset Origem)

Transição 9 - Ativo para Inativo

- Ocorreu alguma exceção de software nesta UCP (cão-de-guarda, access violation, illegal instruction)
- Esta UCP não consegue controlar o barramento, e sabe que a outra UCP está em estado Reserva. Esta condição não é avaliada nos primeiros 2 segundos em estado Ativo

Transição 10 - Ativo para Reserva

- As duas UCPs, por algum motivo, estão no estado Ativo, e este conflito deve ser resolvido. A UCPA chaveará para o estado Reserva caso este conflito dure certo tempo, e a UCPB fará o mesmo após um tempo diferente, que é menor do que o tempo da UCPA. Desta maneira, em caso de conflito, a UCPA tem prioridade para continuar em estado Ativo
- O botão STAND-BY foi pressionado, e esta UCP sabe que a outra UCP encontra-se no estado Reserva. Esta condição não é avaliada nos primeiros 2 segundos em estado Ativo

Transição 11 – Reserva para Não-Configurado

- Esta UCP foi desligada ou reinicializada (Reset a Quente, Reset a Frio ou Reset Origem)
- A outra UCP encontra-se em estado Ativo e o projeto desta UCP é diferente do projeto da UCP Ativa. Além de ir para o estado Não-Configurado, uma solicitação de configuração é feita. Existem casos onde a UCP pode permanecer em Não-Configurado, como quando o archive é enviado para a UCP Ativa, desta maneira um comando manual de sincronização será necessário. Depois que os projetos forem sincronizados, a UCP sai automaticamente do estado Não-Configurado para o estado Inicializando
- A outra UCP está no estado Ativo, e a versão de firmware desta UCP é incompatível com a versão de firmware da UCP Ativa

Transição 12 – Reserva para Inativo

- Ocorreu alguma exceção de software nesta UCP (cão-de-guarda, access violation, illegal instruction)
- O comando para transicionar para Inativo foi executado. Isto é feito tipicamente para executar uma manutenção programada na UCP Não-Ativa. Deve-se evitar fazer manutenções programadas na UCP Reserva, por isso, antes é aconselhável chaveá-lo para Inativo
- A outra UCP encontra-se em estado Ativo. No entanto, o serviço de sincronização de dados redundantes não funcionou corretamente nos últimos três ciclos da MainTask

Transição 13 – Reserva para Ativo

- O estado da outra UCP é desconhecido devido a falhas no link interno de redundância ou o serviço de sincronização de diagnósticos redundantes não funcionou corretamente nos últimos dois ciclos da MainTask
- O estado da outra UCP é conhecido e diferente de Ativo
- Esta UCP não detectou atividades no barramento

Primeiros Instantes em Estado Ativo

Nos primeiros 2 segundos em estado Ativo, diversas transições que normalmente poderiam tirar a UCP do estado Ativo não são avaliadas (ver subseções anteriores que definem transições a partir do estado Ativo). Por exemplo, durante este tempo, não adianta solicitar o comando de "Trocar para Reserva" para tentar fazer a UCP ir para o estado Reserva.

Apenas duas condições permitem que a UCP saia do estado Ativo nos primeiros 2 segundos. Estas condições são as seguintes:

- Esta UCP foi desligada ou reinicializada (Reset a Quente, Reset a Frio ou Reset Origem), causando sua transição para Não-Configurado, ou ocorreu alguma exceção de software nesta UCP (cão-de-guarda, access violation, illegal instruction), causando transição para Inativo

- As duas UCPs, por algum motivo, estão no estado Ativo, e este conflito deve ser resolvido. A UCPA chaveará para o estado Reserva caso este conflito dure certo tempo, e a UCPB fará o mesmo após um tempo diferente, que é menor do que o tempo da UCPA. Desta maneira, em caso de conflito, a UCPA tem prioridade para continuar em estado Ativo

Além disso, nos primeiros instantes que uma UCP assume o estado Ativo, alguns diagnósticos que não são redundantes podem não ser válidos. O método para não considerar estes diagnósticos possivelmente "inválidos" é descrito na seção Leitura de Diagnósticos Não-Redundantes.

Falhas mais Comuns Causadoras de Switchovers Automáticos entre UCPs

Nesta seção, são listadas as falhas mais comuns que, de forma automática, causam um switchover da UCP Ativa para Não-Ativo, e da UCP Reserva para Ativo. Estas falhas disparam um subconjunto daquelas transições examinadas na seção Transições entre Estados de Redundância.

- Falha no circuito interno de alimentação na UCP Ativa
- Falha no acesso ao barramento do bastidor pela UCP Ativa
- Falhas na UCP HX3040 da UCP Ativa, tais como
  - Cão-de-guarda
  - Reinicialização (Reset a Quente, Reset a Frio ou Reset Origem)
  - Falha nas interfaces de barramento em um ou ambos os canais de sincronismo NETA e NETB
- Exceção de software na UCP Ativa (cão-de-guarda de tarefa, erro de acesso em memória, etc.)

Falhas Associadas à Switchovers entre UCPs Gerenciados pelo Usuário

Entre as transições examinadas na seção Transições entre Estados de Redundância, algumas possibilitam que o usuário gerencie switchovers entre as UCPs, devido a falhas que normalmente não geram switchovers de forma automática.

Existem casos muito particulares, que dependem da filosofia de cada cliente. Considere-se um exemplo em que o sistema SCADA perde comunicação com a UCP Ativa.

Alguns clientes poderão preferir um switchover manual, onde o operador executa o comando através do menu no visor da UCP. O switchover provoca a retomada da comunicação com a nova UCP Ativa.

Também seria possível a própria UCP Ativa detectar sua perda de comunicação com o SCADA, e ativar um comando em RedCmdLocal, usando as estruturas de dados RedCmdLocal para transportar um comando equivalente ao comando do menu no visor da UCP. Esta seria uma solução totalmente automática e sem intervenção do operador, que tipicamente seria implementada na POU UserPrg.

Através de estruturas de dados como aquelas citadas na seção Estruturas de Dados de Diagnósticos, Comandos e Usuário, é possível trocar diagnósticos e comandos entre as UCPs via link interno de redundância e, desta maneira, o usuário pode executar gerenciamentos especiais de redundância, para falhas que normalmente não causariam switchovers. Maiores detalhes sobre estas estruturas de dados são fornecidos nas seções:

- Estrutura de Diagnósticos da Redundância
- Comandos da Redundância
- Informações do Usuário Trocados entre UCPA e UCPB

Abaixo é exemplificado como o usuário pode gerenciar falhas e executar um switchover devido a um erro de "link down" (cabo de rede rompido) nas interfaces Ethernet da UCP Ativa (o código deve ser utilizado na POU UserPrg) e também por controle de erros de comunicação de protocolo das portas seriais:

```
//Verifica se NIC Teaming está habilitado ou não.
IF ( ( DG_HX3040.tDetailed.Ethernet.NET[1].szIP = '0.0.0.0') OR (
DG_HX3040.tDetailed.Ethernet.NET[2].szIP = '0.0.0.0')) THEN
```

```
    //NIC Teaming habilitado: erro nas duas NETs para executar switchover.
    IF ( DG_HX3040.tDetailed.Ethernet.NET[1].bLinkDown AND
    DG_HX3040.tDetailed.Ethernet.NET[2].bLinkDown) THEN
    //Coloca a UCP Local em StandBy.
          DG_HX3040_01.RedCmdLoc.bStandbyLocal := TRUE;
    END_IF
    ELSE
    //NIC Teaming desabilitado: erro em uma NET para executar switchover.
    IF ( DG_HX3040.tDetailed.Ethernet.NET[1].bLinkDown OR
    DG_HX3040.tDetailed.Ethernet.NET[2].bLinkDown) THEN
          //Coloca a UCP Local em StandBy.
          DG_HX3040_01.RedCmdLoc.bStandbyLocal := TRUE;
    END_IF
END_IF

-----------------------------------------------------------------

IF ( ( DG_HX3040.tDetailed.Serial.COM[1].byProtocol <> 0) OR (
DG_HX3040.tDetailed.Serial.COM[2].byProtocol <> 0))THEN
    //Se ocorre de comunicação para executar switchover.
    IF MODBUS_Device_REQDG_0001.byStatus.bCommError THEN
          //Coloca a UCP Local em StandBy.
          DG_HX3040_01.RedCmdLoc.bStandbyLocal := TRUE;
    END_IF
END_IF
```

Nota:
Quando duas interfaces formarem um par NIC Teaming, a interface inativa sempre terá o endereço de IP 0.0.0.0. Este não é um endereço de IP válido e não é possível configurar manualmente uma interface com este endereço.

Nota:
O diagnóstico MODBUS_Device_REQDG_0001 usado na lógica anterior pode ter variação de nome conforme o usuário cria estes diagnósticos, aconselha-se o uso de todas as variáveis de diagnostico de cada requisição usado no projeto nesta lógica.

Tolerância a Falhas

O objetivo principal de uma UCP redundante é o aumento da disponibilidade do sistema. A disponibilidade é a razão entre o tempo em que o sistema está funcionando corretamente e o tempo total desde a implantação do sistema. Por exemplo, se um sistema foi implantado há 10 anos, e durante este período esteve parado no tempo total de um ano devido a falhas, então sua disponibilidade foi de apenas 90%. Disponibilidades desta ordem são geralmente inaceitáveis para sistemas críticos, sendo que valores da ordem de 99,99% ou ainda superiores podem ser solicitados nestes sistemas.

Para atingir disponibilidades desta ordem, são necessárias diversas estratégias:

1. Utilização de componentes mais confiáveis (com alto MTBF, ou tempo médio entre falhas), o que contribuirá para o aumento do MTBF do sistema como um todo
2. Utilização de redundância pelo menos para os componentes mais críticos ou componentes com menor MTBF, de tal maneira que a falha de um componente possa ser tolerada sem parar o sistema. Se a redundância for implementada através da duplicação de componentes, será necessário que os dois falhem para que o sistema como um todo fique indisponível
3. Alta cobertura de diagnósticos, em especial de componentes redundantes. A redundância de componentes é pouco útil para o aumento da disponibilidade quando não se pode descobrir que um componente redundante falhou. Neste caso, a primeira falha em um dos componentes ainda não compromete o sistema, mas por permanecer oculta, algum dia acontecerá a segunda falha que comprometerá o sistema, já que a primeira falha ainda não foi reparada. As falhas podem ser

classificadas entre diagnosticáveis e ocultas, sendo altamente desejável que todas as falhas de componentes redundantes sejam diagnosticáveis

4. Também é importante que componentes não redundantes tenham ampla cobertura de diagnósticos, pois muitas vezes o sistema pode continuar funcionando mesmo com a falha de um componente não redundante. O componente pode não estar sendo solicitado. Por exemplo, um relé com contato aberto, que raramente tem sua bobina acionada, não tem sua falha detectada até o momento em que o sistema solicitar seu fechamento
5. Baixo tempo de reparo de componentes não redundantes. A falha de um componente não redundante pode comprometer o sistema, e durante o reparo, o sistema estará indisponível
6. Possibilidade de reparar ou substituir um componente redundante sem parar o sistema. Se esta possibilidade existe, obtém-se um grande aumento de disponibilidade. Do contrário, deve-se programar uma parada do sistema para substituir o componente, e o tempo de reparo conta como tempo indisponível
7. Baixo tempo de reparo de componentes redundantes. A falha de um componente redundante não compromete o sistema, mas durante seu reparo, eventualmente pode ocorrer uma falha em seu par redundante. Por este motivo, é importante que a falha seja reparada brevemente, depois de diagnosticada. Quanto maior o tempo de reparo, maior a probabilidade de acontecer uma segunda falha no componente redundante durante o reparo da primeira falha, o que comprometeria o sistema. Portanto, quanto maior o tempo de reparo, menor a disponibilidade do sistema
8. Programar testes off-line periódicos em componentes para detectar falhas não diagnosticáveis automaticamente pelo sistema. O objetivo é detectar falhas ocultas, especialmente em componentes redundantes ou componentes simples que não estejam sendo solicitados (por exemplo, um relé de segurança). Testes off-line às vezes implicam em paradas no sistema, o que diminui a disponibilidade. Normalmente aproveitam-se ocasiões especiais, tais como paradas programadas de manutenção da planta. Quanto maior o período entre testes off-line, maior o tempo em que uma falha pode permanecer oculta, e portanto maior a probabilidade de uma falha comprometer o sistema, ou seja, menor a disponibilidade do sistema

Estes princípios foram considerados no projeto de UTRs redundantes com UCP HX3040.

As próximas subseções analisam diversos tipos de falhas e como são toleradas ou não toleradas, e se existem switchovers associados às falhas toleradas.

Falhas Simples com Indisponibilidade

Alguns componentes, por não serem duplicados, não toleram nem sequer falha simples sem causar algum tipo de indisponibilidade. Em uma UTR redundante com UCP HX3040, tratam-se dos seguintes componentes:

- Módulos de E/S
- Bastidores (HX9001 ou HX9003)

Quanto à indisponibilidade de um módulo de E/S, deve-se observar que a mesma não se constitui em uma indisponibilidade total do sistema. Ela se constitui em uma indisponibilidade parcial, somente das malhas de controle que utilizam este módulo de E/S.

Embora não haja previsão de redundância de módulos de E/S, a aplicação do usuário pode gerenciá-la em casos especiais. Por exemplo, o usuário pode inserir três módulos de entradas analógicas diferentes, e implementar um esquema de votação entre trios de entradas analógicas, para algum sistema crítico. No entanto, tais soluções, como foi enfatizado, devem ser gerenciadas pelo usuário. Não existe nenhum suporte automático para as mesmas. Tais soluções, em geral, também implicam na redundância de transdutores e atuadores no campo.

Falhas Simples sem Indisponibilidade Causando um Switchover

UCPs HX3040 redundantes toleram falhas simples sem causar indisponibilidade, mas causam switchover.

Falhas Simples sem Indisponibilidade

Fontes de alimentação HX8320 redundantes toleram falhas simples, sem causar indisponibilidade e sem gerar switchover.

Overhead da Redundância

Uma aplicação redundante causa um aumento no tempo de processamento de uma aplicação, quando comparado ao tempo necessário para uma aplicação equivalente não-redundante.

Este tempo adicional deve-se principalmente à execução dos serviços de sincronização cíclicos, descritos na seção Serviços de Sincronização Cíclicos através de Canais de Sincronismo de Redundância, além de um tempo menor para o próprio gerenciamento da redundância (máquina de estados, etc.). O tempo adicional total, devido à redundância (overhead da redundância), é estimado pelo MasterTool e exibido na janela de Mensagens, após compilar o projeto da UCP redundante.

O tempo para transferência dos dados varia conforme a quantidade de dados e segue uma proporção aproximada de 6,4 ms a cada 100Kb de dados. Para o cálculo efetivo do tempo máximo de transferência, é estimado um tempo mínimo de 7ms (constante) e calcula-se o restante usando a quantidade total de dados (incluindo a fila de eventos, de 117.000 bytes). Sendo assim, tipicamente numa aplicação redundante vazia temos o tempo máximo de transferência calculado em 7ms + 6,4ms/Kbyte * 117 Kbytes = 15ms. Este tempo é mostrado pelo Mastertool como "overhead máximo de redundância".

ATENÇÃO: O overhead calculado pelo MasterTool considera uma lista de forçamentos de variáveis redundantes vazia.

O tempo de ciclo da MainTask deve ser ajustado levando em conta o tempo de "overhead máximo de redundância" calculado pelo Mastertool, mais 30% de folga. Ou seja, no caso de 15ms, o tempo de ciclo deverá ser de pelo menos 20ms. Contudo, em aplicações redundantes, o tempo mínimo permitido é de 50ms, devido ao tempo necessário para processamento da fila de eventos no switchover.

ATENÇÃO: Caso a folga de 30% de tempo não seja respeitada, o sistema não garantirá um funcionamento de redundância adequado. O que normalmente ocorrerá é a troca de estado de um dos UCPs para INATIVO, devido às falhas de sincronização dos dados entre os UCPs.

Além disso, cabe ao usuário definir um intervalo para a MainTask que acomode:

- O tempo necessário para executar as POUs principais (NonSkippedPrg e UserPrg). Este tempo normalmente é medido após o desenvolvimento do projeto (descontando o tempo adicional para redundância)
- Alguma folga do ciclo de MainTask, para execução de outras tarefas da UCP (sistema operacional, I/O drivers, etc.). O percentual desta folga pode variar de acordo com o desempenho solicitado destas outras tarefas. Por exemplo, se a comunicação MODBUS com o sistema SCADA precisa alocar muito processamento para atingir um desempenho satisfatório, esta folga deverá ser aumentada
- O tempo necessário para executar as POUs de proteção (NonSkippedProtPrg e UserProtPrg), quando utilizadas. Como a prioridade da tarefa ProtTask, associada a estas POUs, é maior que a prioridade da MainTask, a frequência/intervalo da tarefa ProtTask deve ser levada em consideração. O tempo médio de consumo da ProtTask deverá ser de no máximo 20% do seu tempo de ciclo. Isso equivale a 800us se a mesma estiver trabalhando com 4ms de interval. Caso isso não seja respeitado, o sistema não conseguirá executar a transferência dos dados redundantes. A ProtTask também não deverá ter picos muito maiores que os 20% do tempo de ciclo, pois isso pode inclusive gerar um switchover. Assim, recomenda-se não ter lógicas com

laços que possam divergir e consumir muito tempo. Para maiores informações, consultar a seção Tarefas Cíclicas: MainTask e ProtTask

ATENÇÃO: Dependendo do alinhamento de memória, o número de bytes utilizados no cálculo do overhead de redundância poderá ser maior do que o total de bytes declarados nas variáveis.

Programação de uma UCP Redundante

Wizard para Criação de um Novo Projeto Redundante

Para criar um novo projeto redundante, deve-se utilizar o comando Arquivo / Novo Projeto, e selecionar o Projeto MasterTool Padrão.

Inicialmente, o usuário deve informar o nome que deseja dar ao projeto e o diretório onde deseja armazená-lo, conforme mostra a Figura 6-5:

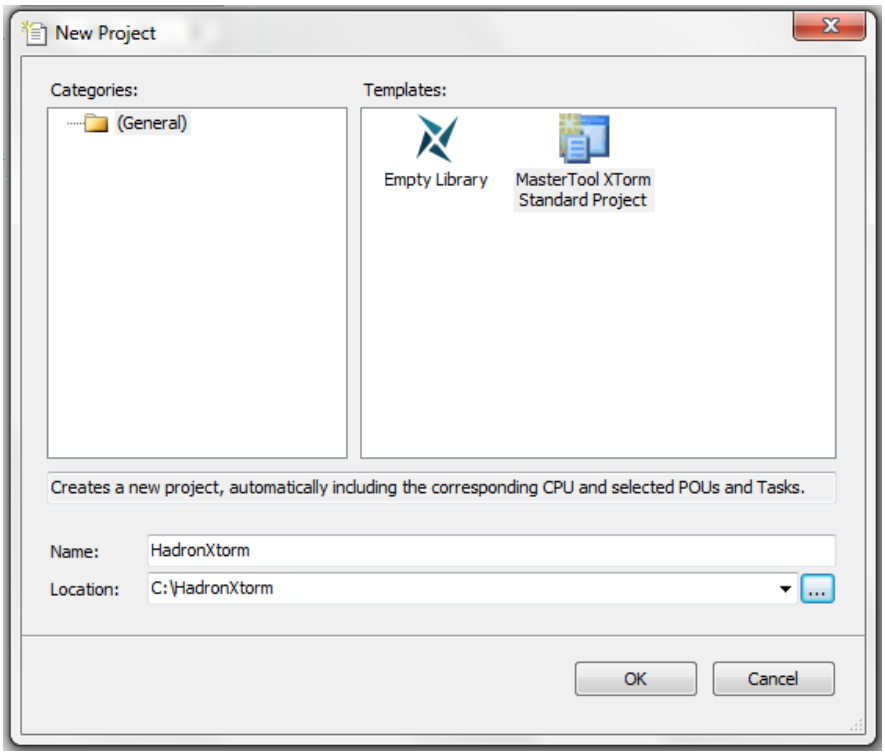


Figura 6-5. Novo Projeto

A seguir, o Wizard que gera o projeto de redundância realiza alguns questionamentos ao usuário quanto à configuração desejada, os quais devem ser respondidos sucessivamente.

O primeiro ponto a ser definido é a configuração inicial de hardware da UTR

- Selecionar o modelo da UCP: como a redundância está implementada somente na HX3040, a mesma deve ser selecionada pelo usuário
- Selecionar o modelo do bastidor: existem duas opções de bastidores disponíveis e a escolha do mesmo dependerá da quantidade de módulos utilizados
- Selecionar o modelo da fonte de alimentação
- Selecionar a configuração da redundância de UCP
- Selecionar a configuração da redundância de fonte de alimentação (essa tipo de configuração para a fonte de alimentação independe da configuração de redundância de UCP)

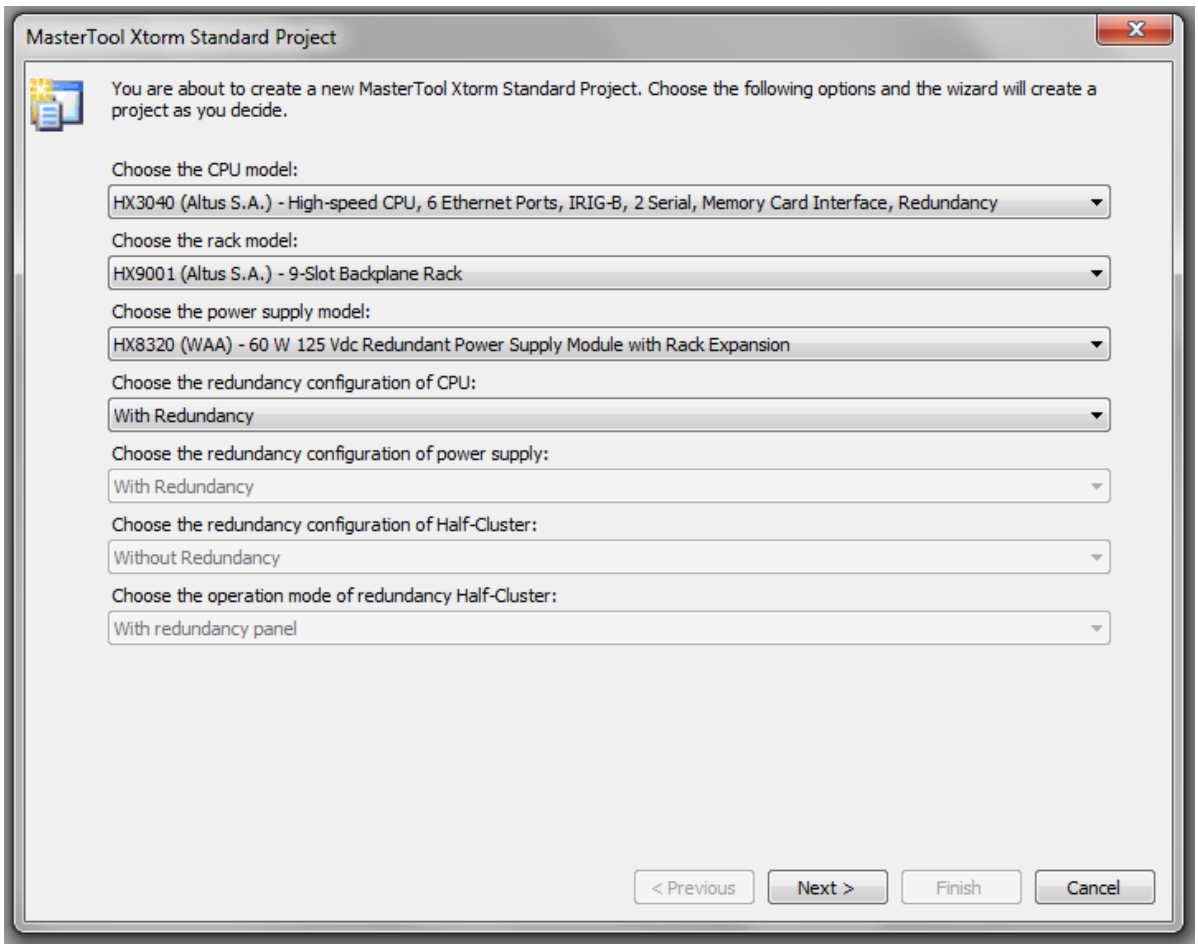


Figura 6-6. Configuração Inicial de Hardware

Após, o usuário deve definir a quantidade e os tipos de E/S utilizadas na aplicação

- Selecionar o número de pontos de entradas digitais
- Selecionar o número de pontos de saídas digitais
- Selecionar o número de pontos de entradas analógicas V/I
- Selecionar o número de pontos de entradas analógicas RTD

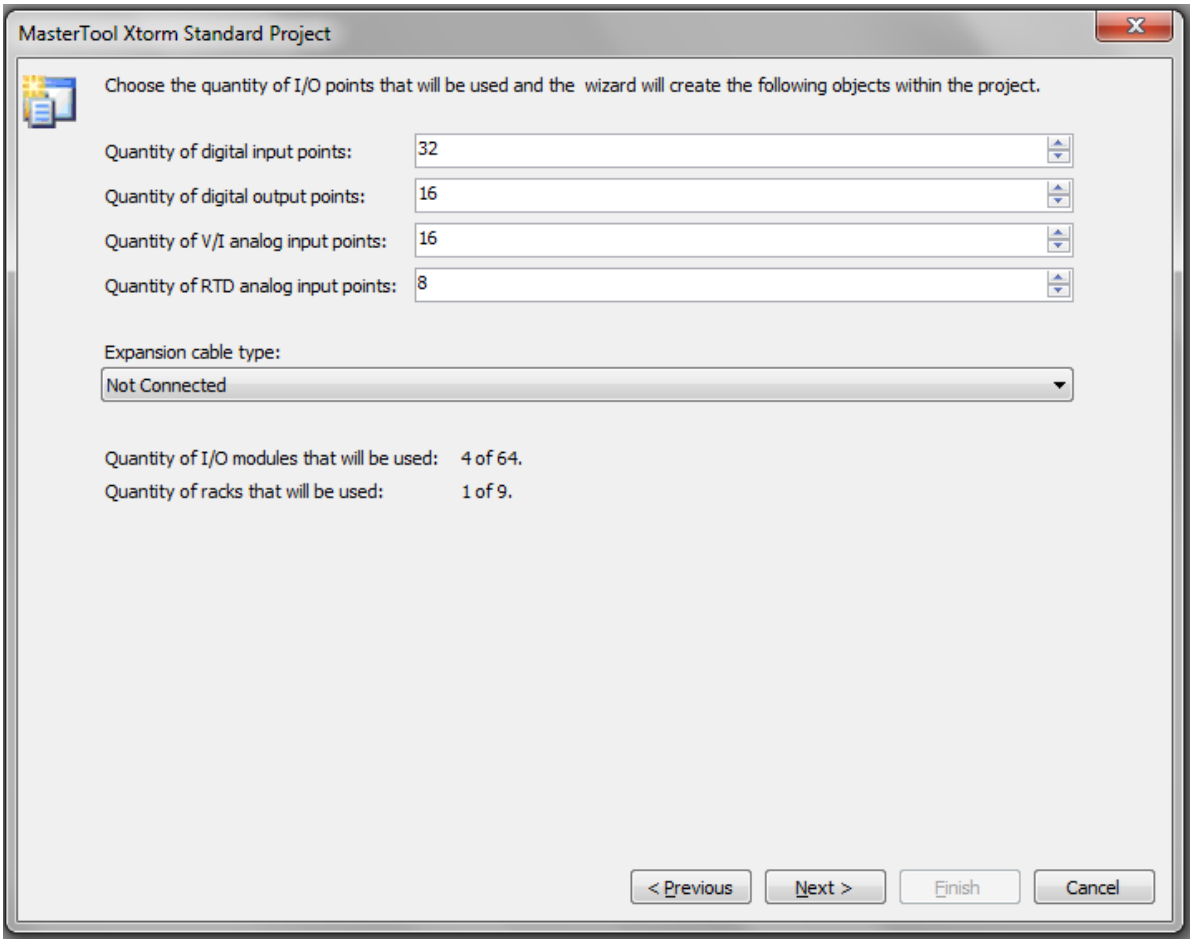


Figura 6-7. Configuração dos pontos de E/S

Então, deve ser selecionado o perfil de projeto e a linguagem padrão para a criação dos programas:

- Selecionar o perfil do projeto: somente é possível utilizar o perfil de projeto “Perfil para UTR” para a redundância, logo a opção de seleção está bloqueada para edição
- Selecionar a linguagem padrão para todos os programas: a linguagem selecionada pelo usuário será a padrão para todas os programas, porém pode optar por utilizar qualquer outra para uma POU específica

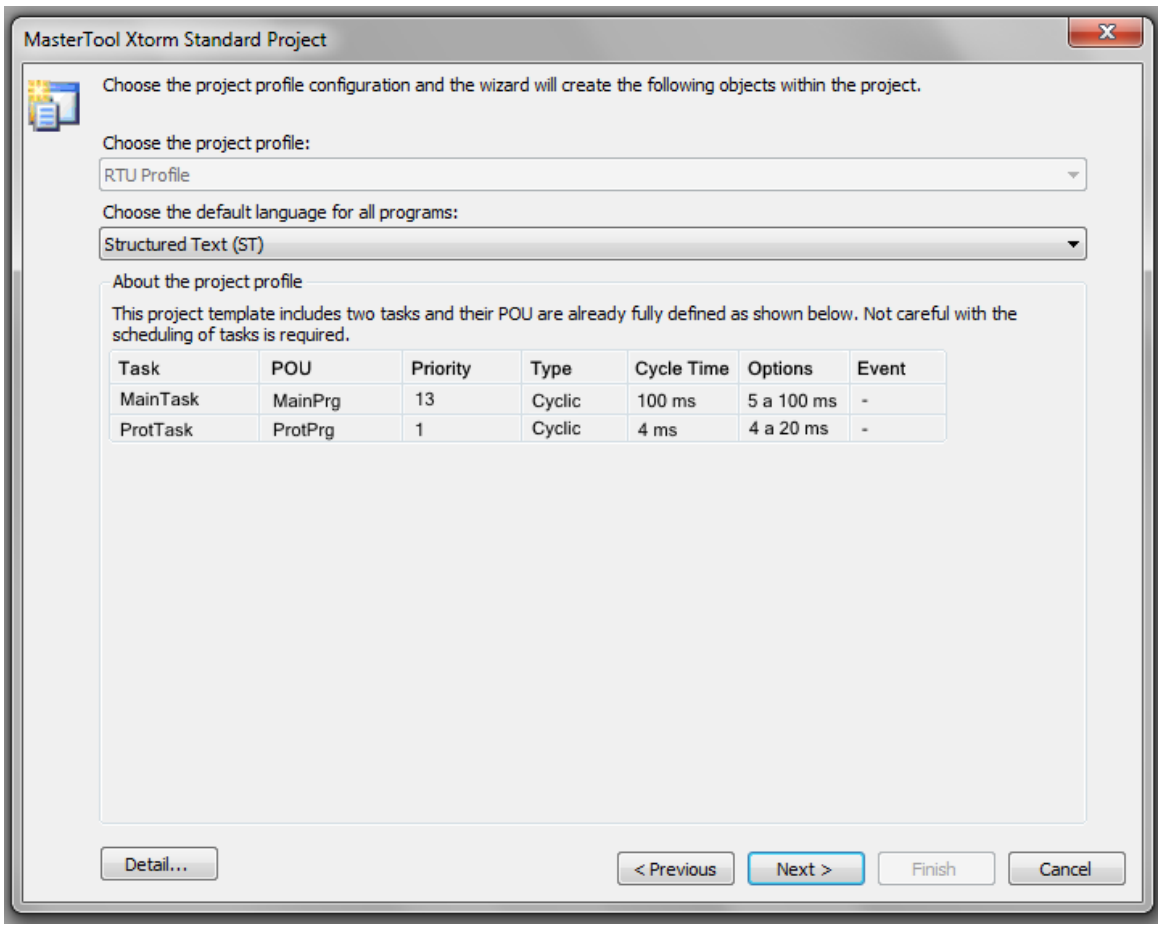


Figura 6-8. Perfil de Projeto e Linguagem Padrão

Para finalizar, o usuário deve selecionar a linguagem de programas comuns e associados à redundância:

- Programas associado à MainTask (MainPrg)
- Programas associados a tarefas cíclicas: deverá ser, obrigatoriamente, em linguagem ST, sendo que o MasterTool desabilita as outras opções
- Programas associados às tarefas principais da redundância

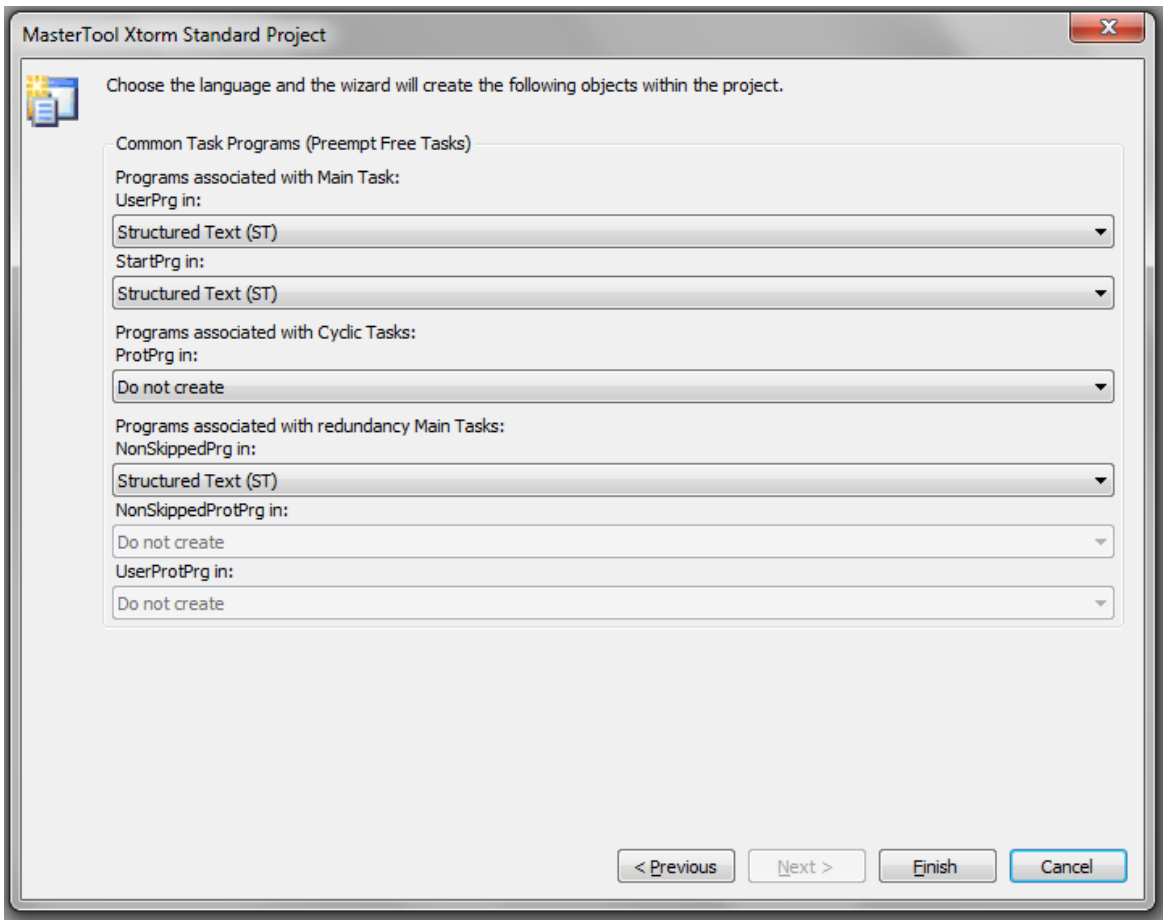


Figura 6-9. Linguagem dos Programas Específicos

ATENÇÃO: As POUs UserPrg e NonSkippedPrg são criadas automaticamente, vazias, na linguagem selecionada nas perguntas anteriores. Em POUs criadas manualmente pelo usuário, poderá ser utilizada qualquer uma das linguagens disponíveis, exceto em POUs redundantes, que não podem ser escritas na linguagem SFC, pois esta utiliza o timer IEC em segundo plano. Para mais informações, ver a seção Limitações na Programação de uma UCP Redundante.

ATENÇÃO: A POU MainPrg será sempre gerada automaticamente em linguagem ST, e não pode ser alterada pelo usuário. Esta POU chama as POUs UserPrg (somente na UCP Ativa) e NonSkippedPrg (em ambas UCPs). De forma equivalente, a POU ProtPrg, quando requisitada pelo usuário para ser criada, é gerada automaticamente em linguagem ST, e não pode ser alterada pelo usuário. Esta POU chama as POUs UserProtPrg (somente na UCP Ativa) e NonSkippedProtPrg (em ambas UCPs).

Depois de obter respostas para as perguntas anteriores, o Wizard gera o projeto inicial, definindo um sistema com a seguinte configuração inicial de hardware:

- Bastidor selecionado
- Fonte de alimentação (posições 0 e 1)
- UCP HX3040 (posições 2 e 3)
- Módulos de E/S (demais posições disponíveis no bastidor)

Configuração do projeto com redundância de UCP

O Wizard sempre é utilizado para gerar a primeira versão de um projeto redundante. Isso garante que a versão inicial do projeto será gerada rápida e corretamente.

No entanto, é possível que algumas modificações sejam necessárias, tal como a inserção de novos módulos de E/S. Os capítulos a seguir mostram como adicionar e configurar os módulos.

Algumas regras e precauções devem ser seguidas para um projeto redundante, conforme descreve a seção seguinte.

Configuração Fixa nas Posições 0 a 3 do Bastidor

Nas posições 0 a 3 do bastidor selecionado sempre devem estar instalados os seguintes módulos:

- Fonte de alimentação redundante (posição 0 e posição 1)
- UCP HX3040 redundante (posição 2 e posição 3)
- Módulos de E/S (posição 4 em diante)

Estes módulos não devem ser removidos do projeto original gerado pelo Wizard.

Qualquer configuração diferente nestas posições resultará em um erro notificado pelo MasterTool na compilação do projeto.

Configurações das Portas Ethernet da UCP HX3040 (NET 1 até NET 6)

Configuração do Endereço de IP

A Figura 6-10 mostra as configurações da porta NET 1 da UCP HX3040. Para abrir esta tela, deve-se dar um duplo clique sobre a NET desejada, abaixo da UCP HX3040 na árvore de dispositivos.

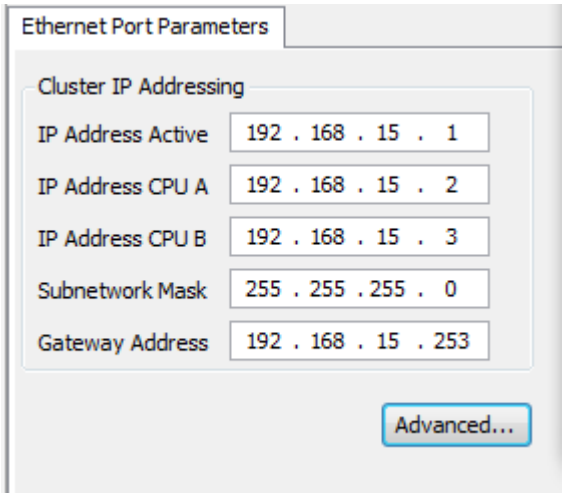


Figura 6-10. Parâmetros da Porta Ethernet NET 1

A seguir devem ser editados os parâmetros básicos para a interface desejada. O endereçamento será de acordo com o método de troca IP Ativo, conforme está descrito em Princípios de Funcionamento > Métodos de Troca de IP > IP Ativo.

ATENÇÃO: Os três endereços IPs (Ativo, da UCPA e da UCPB), bem como o endereço do Gateway, devem pertencer à mesma subrede.

ATENÇÃO:
As NETs ímpares (1, 3 e 5) possuem o botão Avançado, enquanto as NETs pares (2, 4 e 6) não o possuem. Através deste botão é possível configurar redundância de NETs (NIC Teaming) e também selecionar o modo Switch.

Configurações de I/O Drivers

Por definição os I/O Drivers são todos os drivers de comunicação utilizados pela UCP, como Modbus Cliente, Servidor, Mestre, Escravo, drivers Cliente DNP3, Outstation, IEC61850, etc. Estes drivers funcionam de forma diferente em UCPs com redundância.

Para que um I/O Driver esteja rodando, a UCP deve estar em modo RUN e em estado de redundância Ativo. Isto quer dizer que uma UCP Inicializando, Inativa ou em estado Reserva ainda não terá seus I/O Drivers rodando, ou seja, clientes e servidores (MODBUS, DNP3, etc.) estarão parados.

Desta maneira, não é possível utilizar a comunicação de I/O Drivers numa UCP em estado Não-Ativo.

Configurações da MainTask

A tela de configurações associadas à tarefa principal de uma UCP redundante, denominada MainTask, a qual é cíclica, pode ser acessada clicando sobre MainTask na árvore de dispositivos.

Dois parâmetros devem ser ajustados nesta tela:

- Intervalo da MainTask
- Tempo de cão-de-guarda

Diversas considerações devem ser feitas para ajustar adequadamente o intervalo da MainTask:

- O intervalo deve ser suficientemente baixo para controlar o processo efetivamente, observando os tempos de resposta de todos os laços de controle
- O intervalo deve ser suficientemente alto para acomodar, no mínimo, a soma dos dois seguintes tempos:
  - O tempo máximo de execução das POUs NonSkippedPrg e UserPrg, em conjunto
  - O tempo necessário para gerenciar a redundância (overhead da redundância)
- Além disso, o intervalo deve ter uma folga adicional, necessária para que outros processos tenham tempo de ser executados (comunicação Ethernet com sistemas SCADA, etc.)

O MasterTool tem condições de calcular o tempo necessário para gerenciar a redundância (overhead de redundância), depois que o projeto estiver terminado (todas POUs desenvolvidas, e áreas de memória redundantes definidas)

Quanto ao tempo máximo de execução das POUs NonSkippedPrg e UserPrg, é possível medir os tempos depois que estas POUs forem desenvolvidas. Após cada compilação do projeto, o MasterTool soma o overhead de redundância calculado com o parâmetro que informa os tempos das POUs (NonSkippedPrg e UserPrg), e verifica se a folga mínima parametrizada está sendo obedecida

Por exemplo:

- Parâmetros configurados na tela da MainTask
  - intervalo da MainTask: 100 ms
  - Tempo estimado para POUs NonSkippedPrg + UserPrg: 10 ms
  - Folga Mínima: 30%
- Overhead calculado para redundância: 50 ms

Neste caso, o tempo total utilizado é de 60 ms (10 ms + 50 ms), o que consiste em 60% do ciclo da MainTask (100 ms). Desta forma, a folga é de 40%, e portanto a folga mínima de 30% está sendo respeitada.

Programa StartPrg

Nesta POU o usuário pode criar lógicas, laços, iniciar variáveis, etc. que serão executados somente uma única vez no primeiro ciclo de cada uma das UTRs, não sendo chamado novamente durante a execução do projeto. Caso o usuário passe o projeto para o modo Parado, por exemplo em uma manutenção, ao iniciar o projeto para Execução novamente, está POU será executada, bem como em condições de Reset Origin, Reset Cold e Reset Warm.

Programa UserPrg

Nesta POU o usuário deve criar a aplicação principal, responsável pelo controle de seu processo. Esta POU é chamada pela POU principal (MainPrg), sendo executada apenas na UCP Ativa.

O usuário pode também criar POUs adicionais (programa, funções ou bloco funcional), e chamá-las ou instanciá-las dentro da POU UserPrg, para fins de estruturação de seu programa. Também é possível chamar funções e instanciar blocos funcionais definidos em bibliotecas.

Deve-se lembrar que todas as variáveis simbólicas definidas na POU UserPrg, assim como instâncias de blocos funcionais ali definidas, serão variáveis redundantes.

Variáveis simbólicas definidas em POUs adicionais do tipo programa, mesmo que chamadas dentro de UserPrg, só serão variáveis redundantes se o usuário desejar, marcando-a como redundante na configuração do objeto Redundancy Configuration, na árvore de dispositivos do projeto. Por padrão, todas as POUs criadas pelo usuário serão, inicialmente, redundantes.

ATENÇÃO: Variáveis do tipo VAR_TEMP não devem ser utilizadas no programa redundante.

Programa NonSkippedPrg

Esta POU é destinada para controles que devem ser executados em ambos UCPs (UCPA e UCPB), independente do seu estado de redundância. Esta POU é chamada pela POU principal (MainPrg).

Deve-se lembrar que todas as variáveis simbólicas definidas na POU NonSkippedPrg, assim como instâncias de blocos funcionais ali definidas, serão variáveis não-redundantes.

O usuário pode também criar POUs adicionais (programa, funções ou bloco funcional), e chamá-las ou instanciá-las dentro da POU NonSkippedPrg, para fins de estruturação de seu programa. Também é possível chamar funções e instanciar blocos funcionais definidos em bibliotecas.

ATENÇÃO: Quando chamar POUs adicionais do tipo programa dentro de NonSkippedPrg, desmarque as mesmas na janela Redundancy Configuration do MasterTool. Por padrão as variáveis simbólicas declaradas dentro destas POUs serão redundantes e, dentro da NonSkippedPrg, normalmente desejam-se variáveis não redundantes. Normalmente o código de NonSkippedPrg é pequeno, dispensando a chamada de POUs adicionais do tipo programa para sua estruturação, mas, caso a sua estruturação seja necessária, o mais recomendável é utilizar bloco funcional ou funções.

ATENÇÃO: Não se recomenda utilizar os blocos funcionais TOF_RET, TON_RET, TOF e TON no programa NonSkippedPrg. Ver seção Limitações na Programação de uma UCP Redundante.

GVL Disables

Na GVL "Disables" são declaradas as variáveis de desabilitação das requisições MODBUS Mestre/Cliente. Não é obrigatória, mas é recomendada a utilização da geração automática destas variáveis, o que é feito clicando no botão "Gerar Variáveis de Desabilitação" na aba de requisições do dispositivo. Essas variáveis são declaradas com o tipo BOOL e seguem a seguinte estrutura:

Declaração de variável de desabilitação de requisição

```
[Nome do Dispositivo]_DISABLE_[Número da Requisição] : BOOL;
```

Onde:

Nome do dispositivo: Nome que aparece na TreeView para o dispositivo MODBUS.

Número da Requisição: Número da requisição que foi declarada na tabela de requisições do dispositivo MODBUS seguindo a sequência de cima para baixo, começando em 0001.

Exemplo:

Disables

```
VAR_GLOBAL
    MODBUS_Device_DISABLE_0001    : BOOL;
    MODBUS_Device_DISABLE_0002    : BOOL;
    MODBUS_Device_DISABLE_0003    : BOOL;
    MODBUS_Device_1_DISABLE_0001  : BOOL;
    MODBUS_Device_1_DISABLE_0002  : BOOL;
END_VAR
```

A geração automática através do botão "Gerar Variáveis de Desabilitação" apenas cria variáveis, e não remove automaticamente. Desta forma, caso alguma relação seja removida, a sua respectiva variável de desabilitação deve ser removida manualmente.

A GVL "Disables" é editável, portanto as variáveis de desabilitação das requisições podem ser criadas manualmente não precisando seguir o modelo criado pela declaração automática e podem ser usadas as duas maneiras ao mesmo tempo, mas devem sempre ser do tipo BOOL e deve-se tomar o cuidado para não excluir ou alterar as variáveis declaradas automaticamente, pois elas podem estar sendo usadas por algum dispositivo MODBUS. Se a variável for excluída ou alterada será gerado um erro ao compilar o projeto. Para corrigir o nome de uma variável declarada automaticamente, deve-se seguir o modelo exemplificado acima de acordo com o dispositivo e a requisição aos quais pertence.

Enviando um projeto para a UCP e colocando-a em "Execução", através da GVL "Disables" é possível monitorar e alterar os valores das variáveis de desabilitação. Se o valor da variável for TRUE significa que a requisição, à qual a variável pertence, está desabilitada e o inverso é válido para quando o valor da variável for FALSE.

Expression	Type	Value	Prepared value	Comment
MODBUS_Device_DISABLE_0001	BOOL	TRUE		
MODBUS_Device_DISABLE_0002	BOOL	FALSE		
MODBUS_Device_DISABLE_0003	BOOL	TRUE		
MODBUS_Device_1_DISABLE_0001	BOOL	TRUE		
MODBUS_Device_1_DISABLE_0002	BOOL	FALSE		

Figura 6-11. GVL Disable em Modo Online

GVL IOQualities

Na GVL "IOQualities" são declaradas as variáveis de qualidade dos módulos de E/S. Essa GVL não é editável e as variáveis são declaradas automaticamente como arrays do tipo “LibDataTypes.QUALITY" e dimensões de acordo com a quantidade de E/S do módulo ao qual pertence quando este é adicionado ao projeto.

Exemplo:

```
IOQualities
VAR_BLOBAL
    QUALITY_HX1120: ARRAY[0..31]  OF LibDataTypes.QUALITY;
    QUALITY_HX2320: ARRAY[0..15]  OF LibDataTypes.QUALITY;
    QUALITY_HX6000: ARRAY[0..15]  OF LibDataTypes.QUALITY;
    QUALITY_HX6020: ARRAY[0..7]   OF LibDataTypes.QUALITY;
```

```
END_VAR
```

Para os módulos de E/S as variáveis de qualidade comportam-se como mostra a Tabela 6-2. Comportamento de Qualidade de E/S:

Validade	Flags	Descrição
Comportamento comum para todos os módulos de E/S		
VALIDITY_GOOD	-	O módulo está operando normalmente.
VALIDITY_QUESTIONABLE	OLD_DATA & FAILURE	Módulo inoperante, mas o dado já foi lido pelo menos uma vez.
VALIDITY_INVALID	RESTART	Valor inicial.
Comportamento específico para módulos analógicos		
VALIDITY_INVALID	OUT_OF_RANGE	Valor está acima ou abaixo da faixa permitida na entrada do módulo.
	FAILURE	Falha de laço aberto.
	OPERATOR_BLOCKED	O diagnóstico "bInputNotEnable" está setado para a entrada.

Tabela 6-2. Comportamento de Qualidade de E/S

Enviando uma aplicação para a UCP e colocando em "RUN", através da GVL "IOQualities", é possível monitorar os valores das variáveis de qualidade dos módulos que foram adicionados ao projeto.

GVL Module_Diagnostics

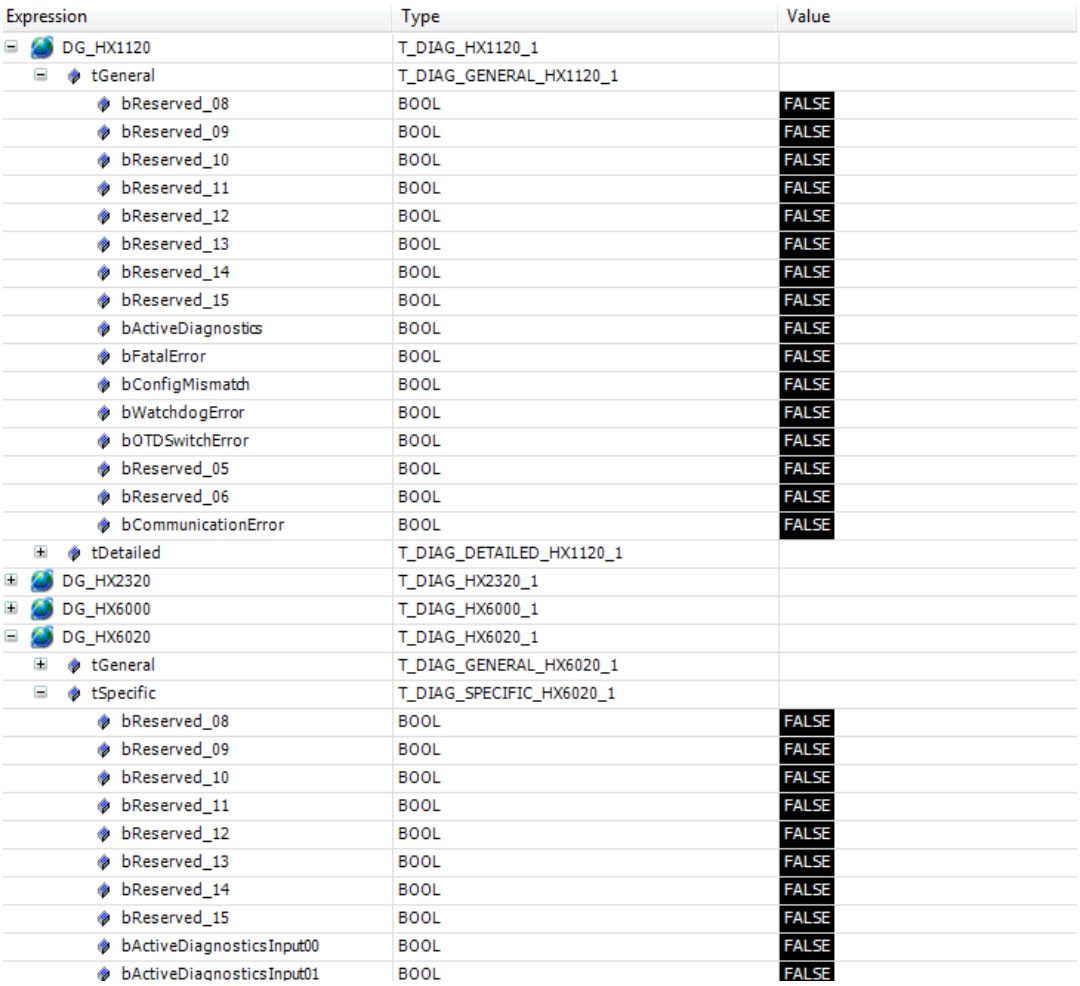
Na GVL "Module_Diagnostics" são declaradas as variáveis de diagnóstico dos módulos utilizados no projeto, exceto UCP e drivers de comunicação. Essa GVL não é editável e as variáveis são declaradas automaticamente com tipo especificado pelo módulo ao qual pertence quando este é adicionado ao projeto.

Exemplo:

Module_Diagnostics

```
VAR_GLOBAL
    DG_HX1120 : T_DIAG_HX1120_1;
    DG_HX2320 : T_DIAG_HX2320_1;
    DG_HX6000 : T_DIAG_HX6000_1;
    DG_HX6020 : T_DIAG_HX6020_1;
    DG_HX8320 : T_DIAG_HX8320_1;
END_VAR
```

Enviando uma aplicação para a UCP HX3040 e passando-a para “Execução”, através da GVL "Module_Diagnostics" é possível monitorar os diagnósticos de cada módulo que foi adicionado ao projeto, como é possível visualizar na Figura 6-12



Expression	Type	Value
DG_HX1120	T_DIAG_HX1120_1	
tGeneral	T_DIAG_GENERAL_HX1120_1	
bReserved_08	BOOL	FALSE
bReserved_09	BOOL	FALSE
bReserved_10	BOOL	FALSE
bReserved_11	BOOL	FALSE
bReserved_12	BOOL	FALSE
bReserved_13	BOOL	FALSE
bReserved_14	BOOL	FALSE
bReserved_15	BOOL	FALSE
bActiveDiagnostics	BOOL	FALSE
bFatalError	BOOL	FALSE
bConfigMismatch	BOOL	FALSE
bWatchdogError	BOOL	FALSE
bOTDSwitchError	BOOL	FALSE
bReserved_05	BOOL	FALSE
bReserved_06	BOOL	FALSE
bCommunicationError	BOOL	FALSE
tDetailed	T_DIAG_DETAILED_HX1120_1	
DG_HX2320	T_DIAG_HX2320_1	
DG_HX6000	T_DIAG_HX6000_1	
DG_HX6020	T_DIAG_HX6020_1	
tGeneral	T_DIAG_GENERAL_HX6020_1	
tSpecific	T_DIAG_SPECIFIC_HX6020_1	
bReserved_08	BOOL	FALSE
bReserved_09	BOOL	FALSE
bReserved_10	BOOL	FALSE
bReserved_11	BOOL	FALSE
bReserved_12	BOOL	FALSE
bReserved_13	BOOL	FALSE
bReserved_14	BOOL	FALSE
bReserved_15	BOOL	FALSE
bActiveDiagnosticsInput00	BOOL	FALSE
bActiveDiagnosticsInput01	BOOL	FALSE

Figura 6-12. GVL Modules_Diagnostic em Modo Online

GVL Qualities

Na GVL "Qualities" são declaradas as variáveis de qualidade dos mapeamentos de variáveis internas, MODBUS Mestre/Cliente e Cliente DNP3. Não é obrigatória, mas é recomendada a utilização da geração automática destas variáveis, o que é feito clicando no botão "Gerar Variáveis de Desabilitação" na aba de requisições do dispositivo. Essas variáveis são declaradas com o tipo "LibDataTypes.QUALITY" e seguem a seguinte estrutura:

Declaração de variável de qualidade de mapeamento

```
[Nome do Dispositivo]_QUALITY_[Número do Mapeamento]:
LibDataTypes.QUALITY;
```

Onde:

Nome do dispositivo: Nome que aparece na TreeView para o dispositivo.

Número do Mapeamento: Número do mapeamento que foi declarado na tabela de mapeamentos do dispositivo seguindo a sequência de cima para baixo, começando em 0001.

Exemplo:

```
Qualities
VAR_GLOBAL
    MODBUS_Device_QUALITY_0001: LibDataTypes.QUALITY;
    MODBUS_Device_QUALITY_0002: LibDataTypes.QUALITY;
    MODBUS_Device_QUALITY_0003: LibDataTypes.QUALITY;
    Outstation_QUALITY_0001: LibDataTypes.QUALITY;
END_VAR
```

A GVL "Quality" é editável, portanto as variáveis de qualidade dos mapeamentos podem ser criadas manualmente não precisando seguir o modelo criado pela declaração automática e podem ser usadas as duas maneiras ao mesmo tempo, mas devem sempre ser do tipo "LibDataTypes.QUALITY" e deve-se tomar o cuidado para não excluir ou alterar as variáveis declaradas automaticamente, pois elas podem estar sendo usadas por algum dispositivo. Se a variável for excluída ou alterada será gerado um erro ao compilar o projeto. Para corrigir o nome de uma variável declarada automaticamente, deve-se seguir o modelo exemplificado acima de acordo com o dispositivo e o mapeamento aos quais pertence.

Para os dispositivos de comunicação as variáveis de qualidade se comportam da seguinte maneira, ver Tabela 4-11.

Enviando uma aplicação para a UCP HX3040 e passando-a para “Execução”, através da GVL "Qualities", é possível monitorar os diagnósticos os valores das variáveis de diagnósticos de mapeamentos dos dispositivos MODBUS Mestre/Cliente e Cliente DNP3, conforme mostra a Figura 6-13.

Expression	Type	Value
MODBUS_Device_QUALITY_0001	Rtu_Standard.QUALITY	
VALIDITY	QUALITY_VALIDITY	VALIDITY_INVALID
FLAGS	QUALITY_FLAGS	
FLAG_OUT_OF_RANGE	BOOL	FALSE
FLAG_INACCURATE	BOOL	FALSE
FLAG_OLD_DATA	BOOL	FALSE
FLAG_FAILURE	BOOL	FALSE
FLAG_OPERATOR_BLOCKED	BOOL	FALSE
FLAG_TEST	BOOL	FALSE
FLAG_RESERVED_0	BOOL	FALSE
FLAG_RESERVED_1	BOOL	FALSE
FLAG_RESTART	BOOL	TRUE
FLAG_COMM_FAIL	BOOL	FALSE
FLAG_REMOTE_SUBSTITUTED	BOOL	FALSE
FLAG_LOCAL_SUBSTITUTED	BOOL	FALSE
FLAG_FILTER	BOOL	FALSE
FLAG_OVERFLOW	BOOL	FALSE
FLAG_REFERENCE_ERROR	BOOL	FALSE
FLAG_INCONSISTENT	BOOL	FALSE
MODBUS_Device_QUALITY_0002	Rtu_Standard.QUALITY	
VALIDITY	QUALITY_VALIDITY	VALIDITY_INVALID
FLAGS	QUALITY_FLAGS	
MODBUS_Device_QUALITY_0003	Rtu_Standard.QUALITY	
VALIDITY	QUALITY_VALIDITY	VALIDITY_INVALID
FLAGS	QUALITY_FLAGS	
Outstation_QUALITY_0001	Rtu_Standard.QUALITY	
VALIDITY	QUALITY_VALIDITY	VALIDITY_QUESTIONABLE
FLAGS	QUALITY_FLAGS	

Figura 6-13. GVL Qualities em Modo Online

GVL ReqDiagnostics

Na GVL "ReqDiagnostics" são declaradas as variáveis de diagnóstico das requisições MODBUS Mestre/Cliente e Cliente DNP3. A declaração destas variáveis segue a seguinte estrutura:

Declaração de variável de diagnóstico de requisição

```
[Nome do Dispositivo]_REQDG_[Número da requisição]: [Tipo da Variável];
```

Onde:

Nome do dispositivo: Nome que aparece na TreeView para o dispositivo.

Número da Requisição: Número da requisição que foi declarada na tabela de requisições do dispositivo seguindo a sequência de cima para baixo, começando em 0001.

Tipo da Variável: "T_DIAG_MODBUS_RTU_MAPPING_1" para MODBUS Master, "T_DIAG_MODBUS_ETH_MAPPING_1" para MODBUS Cliente e "T_DIAG_DNP_CLIENT_REQUEST_1" para Cliente DNP3.

Exemplo:

```
ReqDiagnostics
VAR_GLOBAL
    MODBUS_Device_REQDG_0001      : T_DIAG_MODBUS_RTU_MAPPING_1;
    MODBUS_Device_REQDG_0002      : T_DIAG_MODBUS_RTU_MAPPING_1;
    MODBUS_Device_REQDG_0003      : T_DIAG_MODBUS_RTU_MAPPING_1;
    MODBUS_Device_1_REQDG_0001    : T_DIAG_MODBUS_ETH_MAPPING_1;
    MODBUS_Device_1_REQDG_0002    : T_DIAG_MODBUS_ETH_MAPPING_1;
    Outstation_REQDG_0001         : T_DIAG_DNP_CLIENT_REQUEST_1;
END_VAR
```

A GVL "ReqDiagnostics" é editável, portanto as variáveis de diagnóstico das requisições podem ser criadas manualmente não precisando seguir o modelo criado pela declaração automática e podem ser usadas as duas maneiras ao mesmo tempo, mas as variáveis devem ser sempre do tipo referente ao dispositivo, como exemplificado acima, e deve-se tomar o cuidado para não excluir ou alterar as variáveis declaradas automaticamente, pois elas podem estar sendo utilizadas por um dispositivo. Se a variável for excluída ou alterada será gerado um erro ao compilar o projeto. Para corrigir o nome de uma variável declarada automaticamente, deve-se seguir o modelo exemplificado acima de acordo com o dispositivo e a requisição aos quais pertence.

Enviando uma aplicação para a UCP HX3040 e passando-a para “Execução”, através da GVL "ReqDiagnostics" é possível monitorar os diagnósticos os valores das variáveis de diagnósticos de mapeamentos dos dispositivos MODBUS Mestre/Cliente e Cliente DNP3, conforme mostra a Figura 6-14.

Expression	Type	Value
MODBUS_Device_REQDG_0001	T_DIAG_MODBUS_RTU_MAPPING_1	
byStatus	T_DIAG_MODBUS_RTU_MAPPING_STATUS	
bCommIdle	BOOL	FALSE
bCommExecuting	BOOL	FALSE
bCommPostponed	BOOL	FALSE
bCommDisabled	BOOL	FALSE
bCommOk	BOOL	FALSE
bCommError	BOOL	FALSE
bDiag_6_reserved	BOOL	FALSE
bDiag_7_reserved	BOOL	FALSE
eLastErrorCode	MASTER_ERROR_CODE	NO_ERROR
eLastExceptionCode	MODBUS_EXCEPTION	NO_EXCEPTION
byDiag_3_reserved	BYTE	0
wCommCounter	WORD	0
wCommErrorCounter	WORD	0
MODBUS_Device_REQDG_0002	T_DIAG_MODBUS_RTU_MAPPING_1	
MODBUS_Device_REQDG_0003	T_DIAG_MODBUS_RTU_MAPPING_1	
MODBUS_Device_1_REQDG_0001	T_DIAG_MODBUS_ETH_MAPPING_1	
MODBUS_Device_1_REQDG_0002	T_DIAG_MODBUS_ETH_MAPPING_1	
Outstation_REQDG_0001	T_DIAG_DNP_CLIENT_REQUEST_1	
eConnectionStatus	CONNECTION_STATUS	CONNECTING
eRequestStatus	REQUEST_STATUS	SUCCESS
tIIN	T_DIAG_DNP_CLIENT_IIN_BITS	
ALL_STATIONS	BOOL	FALSE
CLASS_1_EVENTS	BOOL	FALSE
CLASS_2_EVENTS	BOOL	FALSE
CLASS_3_EVENTS	BOOL	FALSE
NEED_TIME	BOOL	FALSE
LOCAL_CONTROL	BOOL	FALSE
DEVICE_TROUBLE	BOOL	FALSE
DEVICE_RESTART	BOOL	FALSE
NO_FUNC_CODE_SUPPORT	BOOL	FALSE
OBJECT_UNKNOWN	BOOL	FALSE
PARAMETER_ERROR	BOOL	FALSE

Figura 6-14. GVL ReqDiagnostics em Modo Online

GVL System_Diagnostics

Na GVL "System_Diagnostics" são declaradas as variáveis de diagnóstico da UCP e dos drivers de comunicação. Essa GVL não é editável e as variáveis são declaradas automaticamente com tipo especificado pelo dispositivo ao qual pertence quando este é adicionado ao projeto.

Exemplo:

System_Diagnostics

```
VAR_GLOBAL
    DG_HX3040                       : T_DIAG_HX3040_1;
    DG_DNP3_Client                  : T_DIAG_DNP_CLIENT_1;
    DG_DNP3_Server                  : T_DIAG_DNP_SERVER_1;
    DG_MODBUS_Symbol_Client         : T_DIAG_MODBUS_ETH_CLIENT_1;
    DG_MODBUS_Symbol_RTU_Master     : T_DIAG_MODBUS_RTU_MASTER_1;
END_VAR
```

Enviando uma aplicação para a UCP HX3040 e passando-a para “Execução”, através da GVL "System_Diagnostics" é possível monitorar os valores das variáveis de diagnósticos da UCP e dos dispositivos de comunicação MODBUS, IEC 6185 Server e DNP3, conforme mostra a Figura 6-15.

Expression	Type	Value
DG_HX3040	T_DIAG_HX3040_1	
tSummarized	T_DIAG_SUMMARIZED_1	
tDetailed	T_DIAG_DETAILED_1	
DG_DNP3_Client	T_DIAG_DNP_CLIENT_1	
DG_MODBUS_Symbol_Client	T_DIAG_MODBUS_ETH_CLIENT_1	
tDiag	T_DIAG_MODBUS_DIAGNOSTICS	
byDiag_1_reserved	BYTE	0
tCommand	T_DIAG_MODBUS_COMMANDS	
bStop	BOOL	FALSE
bRestart	BOOL	FALSE
bResetCounter	BOOL	FALSE
bDiag_19_reserved	BOOL	FALSE
bDiag_20_reserved	BOOL	FALSE
bDiag_21_reserved	BOOL	FALSE
bDiag_22_reserved	BOOL	FALSE
bDiag_23_reserved	BOOL	FALSE
byDiag_3_reserved	BYTE	0
tStat	T_DIAG_MODBUS_ETH_CLIENT_STATS	
DG_MODBUS_Symbol_RTU_Master	T_DIAG_MODBUS_RTU_MASTER_1	
tDiag	T_DIAG_MODBUS_DIAGNOSTICS	
eErrorCode	SERIAL_STATUS	NO_ERROR
tCommand	T_DIAG_MODBUS_COMMANDS	
byDiag_03_reserved	BYTE	0
tStat	T_DIAG_MODBUS_RTU_MASTER_STATS	
wTXRequsts	WORD	0
wRXNormalResponses	WORD	0
wRXExceptionResponses	WORD	0
wRXIllegalResponses	WORD	0
wRXOverrunErrors	WORD	0
wRXIncompleteFrames	WORD	0
wCTSTimeOutErrors	WORD	0
wDiag_18_reserved	WORD	0

Figura 6-15. GVL System_Diagnostics em Modo Online

Configurações da ProtTask

A "ProtTask" é responsável por executar uma única unidade de programação do tipo "Program", cujo nome é "ProtPrg". Este, por sua vez, pode chamar outras unidades de programação e é responsável por executar as rotinas de proteção do projeto. A “ProtTask” é uma tarefa do tipo cíclica e terá sua prioridade fixada como 1

Programa UserProtPrg

Esta POU é executada somente na UCP Ativa e destina-se ao controle do processo do usuário. A "UserProtPrg" é criada somente em projetos com redundância de UCP e possui uma prioridade maior em relação a "UserPrg".

Programa NonSkippedProtPrg

Esta POU é semelhante a "NonSkippedPrg", com a diferença de possuir uma prioridade maior. A "NonSkippedProtPrg" é criada somente em projetos com redundância de UCP.

GVLs com Variáveis Simbólicas Redundantes

O usuário pode criar outras GVLs, diferentes das citadas anteriormente, para declarar variáveis simbólicas redundantes. Para isso, após a criação da GVL, deve-se marcá-la na configuração do objeto Redundancy Configuration, na árvore de dispositivos do projeto. Por padrão, todas as GVLs criadas pelo usuário serão, inicialmente, redundantes.

ATENÇÃO: Por boa prática, é recomendado evitar a utilização da diretiva AT em GVLs que contenham declarações de variáveis simbólicas redundantes, para evitar o mapeamentos de variáveis em áreas não redundantes.

POUs do Tipo Programa com Variáveis Simbólicas Redundantes

O usuário pode declarar variáveis simbólicas redundantes em POUs do tipo programa, com exceção da POU NonSkippedPrg e da NonSkippedProtPrg onde as variáveis simbólicas declaradas são consideradas não-redundantes.

O usuário pode criar outras POUs, diferentes das citadas anteriormente, para declarar variáveis simbólicas redundantes. Para isso, após a criação da POU, deve marcá-la na configuração do objeto Redundancy Configuration, na árvore de dispositivos do projeto. Por padrão, todas as POUs criadas pelo usuário serão, inicialmente, redundantes.

ATENÇÃO: Por boa prática, é recomendado evitar a utilização da diretiva AT em POUs que sejam redundantes, a fim de evitar o mapeamentos de variáveis em áreas não redundantes.

Utilização de Breakpoints em Sistemas Redundantes

Para sistemas redundantes, só é possível utilizar breakpoints na UCP Ativa, com a outra UCP fora do barramento ou em estado Inativo. O Mastertool bloqueará a operação caso a mesma não respeite esta condição.

Limitações na Programação de uma UCP Redundante

Em uma UCP redundante, irão existir algumas limitações quanto à programação. Estas limitações são tratadas nas subseções a seguir.

Limitações em GVLs e POUs Redundantes

Em uma GVL ou em uma POU do tipo programa que sejam redundantes, as seguintes limitações devem ser respeitadas para um correto funcionamento dos UCPs:

- Não utilizar variáveis do tipo VAR_TEMP
- Não misturar tipos de variáveis (VAR, VAR RETAIN, VAR PERSISTENT, etc.), devendo ser utilizado somente um dos tipos em cada GVL ou POU
- Não misturar declaração de variáveis simbólicas com ATs nas GVLs. Devem ser criadas GVLs separadas, declarando em uma as variáveis do tipo AT e em outra as variáveis simbólicas

- Não armazenar o endereço de uma variável em uma variável redundante (fazer de uma variável redundante um ponteiro para um endereço), pois os endereços das variáveis podem ser diferentes na UCPA e na UCPB

Limitações no Programa Não-Redundante (StartPrg, NonSkippedPrg e NonSkippedProtPrg)

Em uma POU do tipo programa que não seja redundante, no caso, as POUs StartPrg, NonSkippedPrg e NonSkippedProtPrg, as seguintes limitações devem ser respeitadas para um correto funcionamento dos UCPs:

- Não podem ser utilizados os blocos funcionais TON e TOF tradicionais, pois os mesmos utilizam o timer IEC. Quando a UCP Reserva entrar em estado ativo (com a outra UCP saindo do estado Ativo), o timer IEC será sincronizado, causando uma descontinuidade no valor do timer. Deve-se optar pelos blocos funcionais TON_NR e TOF_NR disponibilizados na biblioteca LibPlcStandard. Ver capítulo Configuração > Blocos Funcionais > Timer Não-Redundante
- Não podem ser utilizadas POUs do tipo programa escritas na linguagem SFC (Sequenciamento Gráfico de Funções), pois estas utilizam o timer IEC para temporizar as transições
- Não misturar declaração de variáveis simbólicas com ATs nas GVLs. Devem ser criadas GVLs separadas, declarando em uma as variáveis do tipo AT e em outra as variáveis simbólicas

Obtendo o Estado da Redundância de uma UCP

É possível verificar o estado da redundância de uma UCP a partir da Estrutura de Diagnósticos da Redundância:

```
VAR
    eRedStateLocal : REDUNDANCY_STATE;
END_VAR

eRedStateLocal := DG_HX3040_01.RedDgnLoc.RedundancyDiagnostics.eRedState;
```

Deste modo, o usuário poderá fazer o controle de lógicas que dependam do estado da redundância da UCP, o valor recebido será um valor numérico dado em byte, os significados de todos os estados estão descritos na Tabela 6-3. Estados de redundância a seguir:

eRedState	
Não-Configurado	0
Inicializando	2
Reserva	3
Ativo	4
Inativo	5

Tabela 6-3. Estados de redundância

Leitura de Diagnósticos Não-Redundantes

Um projeto redundante, além de apresentar diagnósticos redundantes (Estrutura de Diagnósticos da Redundância e diagnósticos dos módulos do barramento), apresenta também diagnósticos que não são redundantes (diagnósticos específicos da UCP HX3040). Estes diagnósticos não são trocados entre as UCPs. Por exemplo, se no programa o usuário quiser saber o estado do link das portas ethernet da outra UCP, ele deverá fazê-lo utilizando bytes disponíveis na estrutura de dados de usuário dos diagnósticos da redundância. Veja mais em Informações do Usuário Trocados entre UCPA e UCPB.

Carga de Programas em uma UCP Redundante

A seção Programação de uma UCP Redundante tratou de aspectos relativos ao desenvolvimento de um projeto para uma UCP redundante com UCP HX3040.

Nesta seção, discutem-se métodos e etapas para carregar este projeto em uma UCP redundante, considerando diversas situações, tais como:

- Carga do projeto em uma UCP HX3040 nova, retirada da caixa, ou em uma UCP contendo um projeto desconhecido
- Carga on-line de modificações sem interrupção do controle do processo
- Carga off-line de modificações com interrupção do controle do processo, durante uma parada programada do processo

Carga Inicial de um Projeto Redundante

Esta seção descreve os passos necessários para fazer a primeira carga de projeto redundante em uma UCP HX3040. Isto é necessário, por exemplo, para uma UCP nova de fábrica, ou para uma UCP que contenha um projeto desconhecido.

ATENÇÃO: Os passos seguintes só precisam ser executados para uma das duas UCPs (UCPA ou UCPB) que compõem o sistema redundante. Ao se introduzir no barramento a segunda UCP, esta automaticamente irá se sincronizar com a UCP já em funcionamento, caso a mesma esteja em estado Ativo.

Primeiro Passo - Descoberta do Endereço IP para Conexão do MasterTool

O primeiro passo é descobrir o endereço IP do canal NET 1 desta UCP, para conexão ao MasterTool.

Isto deve ser feito através do visor e botão da UCP HX3040, conforme descrito em Menu Informativo e de Configuração da UCP. O menu REDE informa os endereços IPs de todas as portas NET da UCP HX3040, especialmente a da NET1 por ser a única utilizada para comunicação com o MasterTool.

Segundo Passo – Verificar Rede e IP do Computador para Programação

Para que todos os próximos estágios possam ser seguidos, existe a necessidade de que o computador que irá programar e enviar o projeto para a UCP HX3040 esteja na mesma subrede ethernet e que possua um IP diferente e único do que será configurado para a UCP.

Terceiro Passo – Verificar Conflito de Endereços IP

Antes de executar o próximo passo, deve-se ter certeza de que não existe, na rede, outro equipamento com o mesmo endereço IP descoberto nos passos anteriores. O usuário inicialmente deve verificar se o firewall do Windows está desabilitado, conforme seguem as instruções do site oficial da Microsoft http://windows.microsoft.com/pt-br/windows/turn-windows-firewall-on-off#turn-windows-firewall-on-off=windows-7, alguns sistemas operacionais bloqueiam por padrão o eco dessas mensagens e necessitam de um cuidado especial. Feito isso, o IP pode ser descoberto, por exemplo, desconectando a UCP da rede e executando um comando “ping” no seu endereço IP. Como a UCP está desconectada, espera-se que este “ping” falhe. Se o “ping” responder, existe outro equipamento com o mesmo endereço IP.

Caso o endereço IP já esteja em uso por outro equipamento na rede, deve-se executar o próximo passo, e também alguns passos seguintes, utilizando um cabo de crossover para conectar diretamente o PC com o software MasterTool Xtorm à UCP, evitando assim conflitos de endereços IP. Em um dos passos seguintes, ao carregar o projeto na UCP, serão atualizados os endereços IP definitivos da UCP (ver seção Configurações das Portas Ethernet da UCP HX3040).

Quarto Passo – Preparar Conexão do MasterTool (Definir Caminho Ativo)

O quarto passo consiste em dar um duplo clique sobre o Device (HX3040) na árvore de dispositivos, entrar na aba “Configurações de Comunicação”, clicar sobre o Gateway, e pressionar o botão “Mapear Rede” para listar todos os UCPs detectados pelo MasterTool na rede.

Neste momento, espera-se encontrar uma lista de CPs cuja identificação contém o endereço IP descoberto no primeiro passo e o tipo da UCP (HX3040). Caso o usuário tenha trocado anteriormente o nome da UCP na rede, este nome será o visualizado neste momento. A seção Localizando a Rede descreve com maiores detalhes as possíveis identificações que podem ser observadas nesta lista, bem como a seleção e envio do projeto.

Quinto Passo – Carga do Projeto Redundante

Este passo descreve a carga do projeto redundante na UCP. Este projeto deve ser preparado conforme descrito na seção Programação de uma UCP Redundante.

Um projeto redundante básico pode ser preparado seguindo, no mínimo, as seguintes subseções desta seção:

- Wizard para Criação de um Novo Projeto Redundante
- Configurações das Portas Ethernet da UCP HX3040

Obviamente, também é possível fazer um projeto redundante completo e só depois carregá-lo na UCPA e UCPB, por exemplo, caso o hardware destes CPs não esteja disponível durante o desenvolvimento do projeto com o MasterTool.

A primeira carga de um projeto redundante em uma UCP ainda deve ser feita utilizando aquele endereço IP descoberto no primeiro passo deste procedimento, e selecionado nos passos que se sucedem a este procedimento.

A carga do projeto é feita através do menu Comunicação / Login.

Nota:

Ao fim do envio do projeto para uma das UCPs, a outra UCP fará download automaticamente do projeto por canais de sincronismo internos.

ATENÇÃO:
Dentro do projeto desenvolvido com o MasterTool e carregado na UCP neste passo, foram definidos novos endereços IP para a interface NET 1 da UCPA e UCPB (Endereço de IP da UCPA e Endereço de IP da UCPB), assim como um endereço IP para a interface NET 1 da UCP Ativa (Endereço de IP Ativo) – ver seção Configurações das Portas Ethernet da UCP HX3040.
Portanto, depois desta carga inicial, aquele endereço IP descoberto no primeiro passo deste procedimento normalmente não será mais válido. Esta mudança do endereço IP em NET 1 provocará uma perda da conexão do MasterTool com a UCP, que será notificada, somente a NET1 pode ser utilizada para comunicação com a UCP via MasterTool.
Para maiores detalhes sobre como reconectar o MasterTool, ver seção Conexão do MasterTool com uma UCP HX3040 de uma UCP Redundante.

Conexão do MasterTool com uma UCP HX3040 de uma UCP Redundante

Depois de executar o procedimento descrito na seção Carga Inicial de um Projeto Redundante nas duas UCPs (UCPA e UCPB), a conexão ao MasterTool, através da interface NET 1 da UCP HX3040, poderá ser feita através de um dos seguintes endereços:

1. Endereço de IP da UCPA: endereço de NET 1 exclusivo para a UCPA
2. Endereço de IP da UCPB: endereço de NET 1 exclusivo para a UCPB

Independente do estado da UCP, o MasterTool só consegue se conectar ao mesmo utilizando o endereço exclusivo da UCP da NET1, configurado em Endereço de IP da UCPx. Porém, caso a UCP se encontre em estado Ativo, todos os outros serviços poderão se conectar à UCP tanto pelo Endereço de IP da UCPx quanto pelo Endereço de IP Ativo.

Para conectar-se a determinada UCP, em primeiro lugar deve-se dar um duplo clique sobre o Device (HX3040) na árvore de dispositivos, entrar na aba “Configurações de Comunicação”, clicar sobre o

Gateway, e pressionar o botão “Mapear Rede” para listar todos os CPs detectados pelo MasterTool na rede.

Nesta lista, será possível encontrar as seguintes identificações padrão, caso o nome da UCP na rede não tenha sido alterado anteriormente pelo usuário:

3. HX3040_<IP address>_PLCA: identificação relacionada à UCPA. Neste caso, o campo <IP address> deve coincidir com o Endereço de IP da UCPA configurado no projeto.
4. HX3040_<IP address>_PLCB: identificação relacionada à UCPB. Neste caso, o campo <IP address> deve coincidir com o Endereço de IP da UCPB configurado no projeto.

A seguir, deve-se selecionar a UCP desta lista onde o MasterTool deve se conectar, e pressionar o botão “Definir Caminho Ativo”. Posteriormente, ao executar o comando do menu Comunicação / Login, o MasterTool se conectará a esta UCP.

ATENÇÃO:
O MasterTool só consegue se conectar a uma UCP de cada vez. Para conectar-se a diversas UCPs, devem-se abrir múltiplas instâncias do MasterTool, cuidando sempre para abrir o projeto correto em cada instância, por exemplo, conectando-se a UCPA em uma instancia do MasterTool, abrindo o mesmo projeto e conectando-se a UCPB. Alterações não devem ser feitas na condição de múltiplas instancias do mesmo projeto aberto.

Carga de Modificações em um Projeto Redundante

Depois que as duas UCPs (UCPA e UCPB) que compõe o sistema redundante já receberam uma carga inicial, conforme descrito na seção Carga Inicial de um Projeto Redundante, é possível carregar modificações sucessivas do projeto, à medida que tais modificações forem necessárias.

A conexão do MasterTool às UCPs para executar a carga de modificações deve ser feita conforme descreve-se na seção Conexão do MasterTool com uma UCP HX3040 de uma UCP Redundante. Nesta seção explica-se como é possível conectar-se a uma UCP específico (UCPA ou UCPB), ou à UCP Ativa, ou à UCP Não-Ativa.

As modificações sempre devem ser carregadas na UCP Ativa, que se responsabilizará por transferi-las automaticamente para a UCP Não-Ativa, através dos canais de sincronismo entre as UCPs. Portanto, o MasterTool normalmente deve usar o endereço de IP exclusivo da UCP que se encontra no estado Ativo (endereço de IP da UCPx), para se conectar à porta NET 1 da UCP HX3040 da UCP Ativa.

ATENÇÃO:
Não é possível carregar um projeto na UCP Não-Ativa, enquanto a outra UCP estiver no estado Ativo. Só será possível carregar um projeto ou fazer modificações online na UCP Não-Ativa se a outra UCP não estiver no estado Ativo (tipicamente no estado Não-Configurado ou Inativo) ou se estiver ausente do barramento. Da mesma forma, só é possível carregar um projeto na UCP Ativa, caso a outra UCP encontre-se no estado Inativo ou não esteja presente no barramento.

Carga de Modificações Off-Line e On-Line

Modificações de projeto podem ser carregadas off-line ou on-line.

Cargas off-line requerem a parada da UCP onde a modificação deve ser carregada. Por outro lado, cargas on-line permitem que a UCP continue executando sua aplicação enquanto a modificação é carregada.

Alguns tipos de modificações exigem carga off-line, ou seja, não podem ser carregadas on-line na UCP onde o MasterTool está conectado. Veja mais em Carga Off-Line de Modificações com Interrupção do Controle do Processo.

Modificações que Demandam Carga Off-Line com Interrupção do Controle do Processo

As seguintes modificações em um projeto farão com que o mesmo se torne impossível de ser carregado num sistema redundante sem a interrupção do controle do processo:

- Adicionar ou remover dispositivos na árvore de dispositivos, por exemplo:
  - Módulos de E/S
  - Instâncias de protocolos de comunicação, como MODBUS, DNP3, etc.
- Modificar parâmetros dentro de dispositivos existentes na árvore de dispositivos, por exemplo:
  - Endereços IP e outros parâmetros de interfaces Ethernet
  - Parâmetros de módulos de E/S
  - Parâmetros de protocolos instanciados, seus mapeamentos, etc.
- Modificações nas configurações das tarefas

Modificações que Permitem Carga On-Line

Em princípio, as modificações não citadas na seção Modificações que Demandam Carga Off-Line com Interrupção do Controle do Processo, permitem carga on-line.

Mesmo assim, a seguir serão citadas as principais modificações que permitem carga on-line na UCP onde o MasterTool está conectado. As modificações citadas abaixo valem para variáveis, POUs e GVLs, redundantes ou não.

- Adicionar POUs do tipo programa, desde que estas POUs não precisem ser associadas a alguma tarefa
- Remover POUs do tipo programa, desde que estas POUs não estejam associadas a alguma tarefa
- Adicionar ou remover POUs do tipo função ou bloco funcional
- Modificar o código de quaisquer tipos de POU (programa, função ou bloco funcional)
- Adicionar ou remover variáveis simbólicas em quaisquer tipos de POU (programa, função ou bloco funcional, sendo elas redundantes ou não)
- Adicionar ou remover instâncias de bloco funcional em POUs do tipo programa ou bloco funcional
- Adicionar ou remover GVLs
- Adicionar ou remover variáveis simbólicas ou instâncias de bloco funcional em GVLs

Existem ainda dois métodos de cargas on-line, descritos a seguir, um para compilação e envio da Aplicação de Inicialização feito manualmente e o outro automaticamente ao fim do envio da carga On-line. Para maiores informações o Manual de Utilização do MasterTool IEC XE MT8500 deve ser consultado MU299048.

Carga On-Line de Modificações

Na seção Carga de Modificações Off-Line e On-Line, foram descritas modificações que demandam carga off-line e aquelas que permitem carga on-line.

Uma carga on-line deve ser feita conectando o MasterTool ao canal NET 1 da UCP Ativa, utilizando seu endereço exclusivo de IP (IP da UCPA ou IP da UCPB). É necessário trocar o estado de redundância da UCP Reserva para Inativo antes de enviar a modificação On-Line para a UCP Ativa, caso contrário o Mastertool bloqueará a operação. Após o envio das modificações, é criada automaticamente a aplicação de inicialização na memória não volátil da UCP, e não se corre o risco de perder as modificações no caso da UCP ser desligada ou no caso de falta de energia.

Depois de finalizado o processo de carga On-Line, pode-se tirar a outra UCP do estado Inativo, por exemplo, usando o comando de “Trocar para Reserva” no menu do seu visor. Isso provocará a transição da UCP para o estado Não-Configurado. A UCP permanecerá no estado Não-Configurado até que o processo automático de sincronização de projetos termine. Depois disso, a UCP passa para o estado Inicializando e depois voltará para o estado Reserva. ou voltar para o estado Inativo se ocorrer alguma falha.

Carga Off-Line de Modificações com Interrupção do Controle do Processo

Nesta seção, define-se o procedimento para executar uma carga off-line que interromperá o controle do processo. Tal situação é aceitável em determinados tipos de processos e durante paradas programadas dos processos.

Uma carga off-line deste tipo deve ser feita conectando o MasterTool ao canal NET 1 da UCP Ativa, utilizando o endereço exclusivo da UCP em estado Ativo (Endereço de IP da UCPx). Antes de iniciar uma carga off-line na UCP Ativa, é necessário passar a UCP reserva para o estado Inativo. Se isto não for feito, o Mastertool não permitirá que a modificação seja aplicada.

Para executar a carga off-line, exige-se então que conscientemente a UCP reserva seja colocada para o estado Inativo, impedindo então que uma carga acidental de programa seja executada.

No momento que carga off-line se iniciar a UCP Ativa irá para o estado Não-Configurado.

Quando a carga off-line terminar, é possível reiniciar a execução do programa na UCP onde a aplicação foi carregada (colocar no modo RUN novamente). Depois de alguns segundos, esta UCP reassume o estado Ativo.

Depois que esta UCP reassume o estado Ativo, pode-se tirar a outra UCP do estado Inativo, por exemplo, usando o comando de "Trocar para Reserva" no menu do seu visor. Isso provocará a transição da UCP para o estado Não-Configurado. Esta UCP permanecerá no estado Não-Configurado até que o processo automático de sincronização de projetos termine. Depois disso, esta UCP passa para o estado Inicializando e depois voltará para o estado Reserva.

Manutenção de UTRs Redundantes

Mensagens de Advertência do MasterTool

Quando o Mastertool for efetuar uma conexão ou já estiver conectado a uma UCP com projeto redundante, algumas mensagens de advertência especiais poderão ocorrer, conforme se descreve nas próximas subseções.

Bloqueio Antes de Comandos que Podem Parar a UCP Ativa

Alguns comandos, como os seguintes, podem parar uma UCP:

- Carga offline após Comunicação / Login
- Depurar / Parar
- Depurar / Novo Breakpoint
- Comunicação / Reset (quente, frio, origem)

Estes comandos só serão disparados caso a UCP Ativa esteja operando com a redundância desarmada, ou seja, a outra UCP deve estar no estado Inativo (ou ausente do barramento). Caso contrário, uma mensagem de bloqueio será mostrada:

"O Login não pode ser realizado, pois o projeto atual é diferente do projeto em uso na UCP. Caso realmente deseje executar esta operação, configure a outra UCP para o estado inativo e execute novamente este comando"

Bloqueios de Operações na UCP Não-Ativa

Em algumas circunstâncias, certas operações são bloqueadas na UCP Não-Ativa. Caso a outra UCP esteja no estado Ativo, o que é o usual no caso da redundância, certas operações serão bloqueadas na UCP Não-Ativa:

- Carga offline e online
- Troca de estado de aplicação (RUN/STOP)
- Depurar / Parar
- Depurar / Novo Breakpoint

Quando houver uma tentativa de execução de um comando deste tipo, o MasterTool emite a seguinte mensagem:

“ Esta operação só poderá ser executada na outra UCP, porque ela está controlando o processo.”

Por outro lado, é possível logar-se na UCP Não-Ativa para monitorar ou forçar variáveis não-redundantes.

Interação com a Redundância Através do Visor Gráfico da UCP HX3040

O estado da redundância e operações disponíveis no menu REDUNDANCIA podem ser acessadas no visor da UCP HX3040.

Estado de Redundância da UCP

O estado de redundância da UCP, descritos em Estados de uma UCP Redundante, é visto nos três caracteres iniciais da segunda linha da tela principal, conforme mostra o capítulo Manutenção – Visor Gráfico. A tela é apresentada após a inicialização da UCP, e volta a ser apresentada alguns segundos depois de terminada qualquer navegação (sem apertar o botão da UCP HX3040).

Telas Abaixo do Menu Redundancia

Existe um menu denominado REDUNDANCIA, abaixo do qual existem algumas telas. A descrição e o acesso às telas de redundância estão disponíveis no capítulo Menu Informativo e de Configuração da UCP.

Estrutura de Diagnósticos da Redundância

A área de diagnósticos da redundância corresponde a variável simbólica DG_HX3040_01, alocada automaticamente pelo Mastertool na GVL System_Diagnostics.

- Esta área é dividida em outras seis estruturas de dados:
  - RedDgnLoc: contém diagnósticos de redundância da UCP local (onde o MasterTool está conectado), como por exemplo, o estado da redundância da UCP. Essa seção é descrita em Diagnósticos da Redundância
  - RedDgnRem: é uma cópia de RedDgnLoc da outra UCP, recebida via canais de sincronismo. Desta forma, a UCP local tem acesso aos diagnósticos da UCP remota. Essa seção é descrita em Diagnósticos da Redundância
  - RedCmdLoc: contém comandos de redundância gerados nesta UCP (local), por exemplo, através de escritas a partir de um sistema SCADA, ou gerados em POUs desta UCP (ex: UserPrg ou NonSkippedPrg). Essa seção é descrita em Comandos da Redundância
  - RedCmdRem: trata-se de uma cópia de RedCmdLoc da outra UCP (remota), recebida via canais de sincronismo. Essa seção é descrita em Comandos da Redundância
  - RedUsrLoc: área de memória de 128 bytes da UCP local, que será transferida para a estrutura de dados RedUsrRem da UCP remota. É utilizada para que o usuário troque informações entre UCPA e UCPB. Essa seção é descrita em Informações do Usuário Trocados entre UCPA e UCPB
  - RedUsrRem: área de memória de 128 bytes na UCP local, que corresponde à estrutura de dados RedUsrLoc da UCP remota. É utilizada para que o usuário troque informações entre UCPA e UCPB. Essa seção é descrita em Informações do Usuário Trocados entre UCPA e UCPB.

É importante ressaltar que as estruturas de diagnósticos da redundância da UCP remota são atualizadas somente quando ocorre uma sincronização de dados com sucesso. Portanto, enquanto uma nova sincronização não ocorrer, os diagnósticos irão permanecer com o valor congelado, correspondente à última troca de dados realizada com sucesso.

Além disso, as estruturas da UCP remota são somente para leitura, isto é, valores escritos nestas estruturas serão sobrescritos, sem serem considerados, na próxima sincronização de dados. Sendo

assim, não é possível utilizar a estrutura RedCmdRem para executar um comando na UCP remota. A estrutura utilizada para executar comandos, a ser escrita, deve ser sempre RedCmdLoc.

Diagnósticos da Redundância

Os diagnósticos da Redundância podem ter várias utilidades. Entre elas:

- Podem ser consultados para verificar a existência de algum problema que precisa ser sanado.
- Podem ser utilizados para consultar os status do sistema, bem como a tomada de ações em casa de falhas ou de variações de valores de diagnósticos. Os principais eventos de redundância relacionados a diagnósticos do sistema são apresentados como eventos nos logs de sistema. Consultando a sequência histórica de tais eventos pode-se descobrir, por exemplo, a causa de um switchover.

ATENÇÃO:
O diagnóstico DG_HX3040_01.RedDgnLoc.sGeneral_Diag.bExchangeSync (definido logo a seguir) deve ser testado para verificar se a estrutura de dados RedDgnRem foi lida com sucesso da UCP remota no último ciclo da MainTask. Caso o valor deste diagnóstico seja FALSE, isso significa que a estrutura de dados RedDgnRem não foi lida com sucesso da UCP remota e, portanto, os valores de RedDgnRem podem ser inválidos ou estar desatualizados.

Como RedDgnRem é uma cópia de RedDgnLoc da outra UCP, as duas estruturas têm o mesmo formato. Estas ainda são divididas em quatro subestruturas:

- RedundancyDiagnostics: diagnósticos gerais da redundância
- SyncLinkDiags[1]: diagnósticos do canal de sincronismo A
- SyncLinkDiags[2]: diagnósticos do canal de sincronismo B
- SyncLinkStatistics: estatísticas comuns para os canais de sincronismo A e B, para contagem de sucessos e falhas dos serviços de sincronização

A subestrutura " RedundancyDiagnostics" possui os seguintes campos para diagnósticos gerais da redundância:

.RedundancyDiagnostics.*	Tipo	Descrição
bConfigDone	BOOL	TRUE – O processo de configuração, executado no estado Não-Configurado, terminou.
		FALSE – O processo de configuração, executado no estado Não-Configurado, ainda não terminou ou não foi executado.
bConfigError	BOOL	TRUE – O processo de configuração, executado no estado Não-Configurado, terminou com erros. Trata-se de um erro de sistema, normalmente não esperado. Entre em contato com o suporte da Altus para reportá-lo. Informe também o valor do diagnóstico ConfigErrorCode para o suporte da Altus.
		FALSE – O processo de configuração ocorreu com sucesso ou não foi realizado.
bTooManyRedAreas	BOOL	TRUE – O número de áreas redundantes excedeu o número máximo permitido. Trata-se de um erro de sistema, normalmente não esperado. Entre em contato com o suporte da Altus para reportá-lo.
		FALSE – O número de áreas redundantes está dentro do esperado.
bTemporaryBufferTooSmall	BOOL	TRUE – Estrutura de dados intermediária com tamanho insuficiente. Trata-se de um erro de sistema, normalmente não esperado. Entre em contato com o suporte da Altus para reportá-lo.
		FALSE – O tamanho da estrutura de dados está dentro do esperado.
bIncompatibleApplication	BOOL	TRUE – A aplicação não é compatível entre os dois CPs. Foi efetuado o download de uma nova aplicação para um dos UCP com uma das seguintes alterações: Modificação da área de dados redundantes; Modificação de variáveis simbólicas redundantes. Enquanto este diagnóstico estiver ligado, um dos CPs ficará em estado Inativo até que a mesma aplicação esteja presente nos dois CPs. Isto implica em recarregar a aplicação antiga nos dois CPs ou atualizar os dois CPs com a nova aplicação. Para mais informações sobre como proceder, consultar capítulo Carga de Programas em uma UCP Redundante.
		FALSE – A aplicação corrente em ambos UCPs é compatível, ou seja, a mesma.
bIncompatibleFirmware	BOOL	TRUE – Esta UCP foi para o estado Não-Configurado pois sua versão de firmware é incompatível com a versão de firmware da UCP Ativa.

		FALSE – A versão de firmware da UCP Ativa é compatível com a versão de firmware da UCP Não-Ativa.
bExchangeSync	BOOL	TRUE – O serviço de sincronização Troca de Diagnósticos e Comandos foi executado com sucesso neste ciclo da MainTask.
		FALSE – A estrutura RedDgnRem tem valores obsoletos ou inválidos, pois não foi lida da outra UCP (remota) neste ciclo da MainTask.
bRedDataSync	BOOL	TRUE – O serviço Sincronização de Dados Redundantes foi executado com sucesso neste ciclo da MainTask.
		FALSE – O serviço Sincronização de Dados Redundantes não foi executado com sucesso neste ciclo da MainTask.
bApplicationProjectDiff	BOOL	TRUE – O projeto aplicativo desta UCP é diferente do presente na outra UCP. UCP Ativa.
		FALSE – O projeto aplicativo deste UCP é igual ao da outra UCP.
bProjectArchiveDiff	BOOL	TRUE – O Project archive desta UCP é diferente do presente na outra UCP.
		FALSE – O Project archive desta UCP é igual ao da outra UCP.
bOnlineChangeApply	BOOL	TRUE – Foi realizada alguma alteração online na aplicação e esta ainda não foi sincronizada com a UCP reserva.
		FALSE – Não foram realizadas alterações online na aplicação ou estas já foram sincronizadas com a UCP reserva.
bBusError	BOOL	TRUE – Detectada alguma falha no acesso ao barramento.
		FALSE – Não há falha de acesso ao barramento.
bBusIdle	BOOL	TRUE – Não há atividade de barramento detectada pela UCP (trabalhando em modo passivo)
		FALSE – Há atividade de barramento detectada pela UCP.
eCPU_ID	ENUM (BYTE)	Este diagnóstico informa a identificação deste UCP: - 0 = não redundante - 2 = UCPA - 3 = UCPB Trata-se de uma cópia da identificação da UCP, conforme descrito na seção Identificação de uma UCP HX3040.
eRedState	ENUM (BYTE)	Informa o estado de redundância desta UCP: - Não-Configurado = 0 - Inicializando = 2 - Reserva = 3 - Ativo = 4 - Inativo = 5
ePreviousRedState	ENUM (BYTE)	Valor que o diagnóstico RedState possuía antes da última transição de estados.
eStateChangeReason	ENUM (BYTE)	Código numérico que indica a razão pela qual o houve a troca de estado da redundância, essas informações estão disponíveis na Tabela 6-7.
eAppState	ENUM (BYTE)	Modo da aplicação: - Desconhecido = 0 - Run = 1 - Stop = 2 - Breakpoint = 3
wRedStateDuration	WORD	Mede há quanto tempo (milissegundos) o estado de redundância atual foi assumido. Este tempo para de incrementar quanto atinge 65535 ms.
dwApplicationCRC	DWORD	CRC de 32 bits do projeto aplicativo, utilizado para detectar diferenças entre os projetos aplicativos das duas UCPs.
dwArchiveCRC	DWORD	CRC de 32 bits do Project archive, utilizado para detectar diferenças entre os Project archives das duas UCPs.
dwFirmwareVersion	DWORD	Versão de firmware desta UCP, utilizada para verificar compatibilidade entre firmware das duas UCPs.
dwIECTimer	dwIECTimer	A sincronização do IEC Timer é necessária para operação bump-less de alguns blocos de função como TON e TOF. Através deste diagnóstico o IEC Timer da UCP Ativa é recebido e atualizado na UCP Não-Ativa, desde que o serviço Troca de Diagnósticos e Comandos tenha sido executado com sucesso. Sua contagem inicia em 0 e incrementa até 4294967295. Após estouro de contagem, reinicia com valor 0.
wCycleCounter	WORD	Contador de 16 bits utilizado como informação auxiliar de sequência nos Logs de Eventos da Redundância. Na UCP Ativa, é incrementado a cada ciclo da MainTask. Na UCP Não-Ativa, recebe uma cópia do valor existente na UCP Ativa, desde que o serviço Troca de Diagnósticos e Comandos tenha sido executado com sucesso. Sua contagem inicia em 0 e incrementa até 65535. Após estouro de contagem, reinicia com valor 0.

Tabela 6-4. Diagnósticos Gerais da Redundância

Nota:

Visualização das Estruturas de Diagnóstico: as estruturas de diagnóstico adicionadas ao projeto podem ser visualizadas acessando o item "Library Manager" na treeview da janela do MasterTool Xtorm. Com isso, é possível visualizar todos os datatypes definidos na estrutura.

A subestrutura "SyncLinkDiags" é um array de dois elementos e possui os seguintes campos para diagnósticos dos dois canais de sincronismo (array 1 e 2):

.SyncLinkDiags[1..2].*	Tipo	Descrição
bGeneralFailure	BOOL	TRUE – O canal de sincronismo possui algum tipo de falha. Os 3 próximos diagnósticos indicarão a falha específica.
		FALSE – O canal de sincronismo está em correto funcionamento.
bInternalFailure	BOOL	TRUE – A falha detectada tem sua causa localizada dentro desta UCP. Tais falhas são tratadas de forma diferenciada.
		FALSE – Nenhuma falha interna.
bLinkDownFailure	BOOL	TRUE – Não há link entre as UCPs.
		FALSE – O link está operacional.
bTimeoutFailure	BOOL	TRUE – Esta falha é reportada caso um serviço de sincronização não tenha terminado com sucesso até um time-out especificado, e não tenham sido encontradas falhas do tipo bInternalFailure ou bLinkDownFailure que justificassem isso.
		FALSE – Não houve time-out.

Tabela 6-5. Diagnósticos Específicos dos links de sincronismo

A subestrutura "SyncLinkStatistics" contém estatísticas de falhas e sucessos dos serviços. As estatísticas das UCPs local e remota podem ser reiniciadas através dos comandos:

```
//UCP Local
DG_HX3040_01.RedCmdLoc.bResetNETStatisticsLocal := TRUE;
//UCP Remoto
DG_HX3040_01.RedCmdLoc.bResetNETStatisticsRemote := TRUE;
```

A subestrutura contém os seguintes contadores:

.SyncLinkStatistics.*	Tipo	Descrição
wSuccessExchDgCmdSync	WORD	Contagem de sucessos do serviço Troca de Diagnósticos e Comandos. Sua contagem inicia em com o valor 0, incrementa até 65535, e então reinicia novamente com o valor 0 quando seu valor limite é ultrapassado.
wFailedExchDgCmdSync	WORD	Contagem de falhas do serviço Troca de Diagnósticos e Comandos. Sua contagem inicia em com o valor 0, incrementa até 65535, e então reinicia novamente com o valor 0 quando seu valor limite é ultrapassado.
wSuccessRedDataSync	WORD	Contagem de sucessos do serviço Sincronização de Dados Redundantes. Sua contagem inicia em com o valor 0, incrementa até 65535, e então reinicia novamente com o valor 0 quando seu valor limite é ultrapassado.
wFailedRedDataSync	WORD	Contagem de falhas do serviço Sincronização de Dados Redundantes. Sua contagem inicia em com o valor 0, incrementa até 65535, e então reinicia novamente com o valor 0 quando seu valor limite é ultrapassado.

Tabela 6-6. Diagnósticos Específicos da Interface

A Tabela 6-7 mostra todas as mensagens possíveis de serem apresentadas para os estados e trocas de redundância, acessíveis ao usuário por meio do diagnóstico "DG_HX3040_01.RedDgnLoc.RedundancyDiagnostics.eStateChangeReason.". As mensagens apresentadas na descrição são idênticas às mensagens que serão apresentadas no Log. Para mais informações sobre Log, consulte os Logs de Eventos da Redundância, para mais informações sobre os estados da redundância ver o capítulo de Transições entre Estados de Redundância.

String do Estado	Valor do Estado	Mensagem apresentada via Log	Descrição

INIT_APP	0	Redundancy initialization started	Processo de redundância foi iniciado.
INIT_DONE	1	Redundancy initialization done	Processo de redundância concluído.
INIT_REDUNDANT_DATA_OVER_MAXIMUM	2	Error: Redundant data is too big	Área destinada à redundância excede o limite máximo.
INIT_CLUCONF_ERROR	3	CPU is not inserted in the right position for a redundant setup	A UCP não esta inserida na posição correta para uma configuração redundante.
NCF_LOCAL_CPU_USER_CMD	100	User state change requested via local diagnostic command bit	Não Configurado para outros estados, comando de troca de estado via bit local de diagnostico.
NCF_REMOTE_CPU_USER_CMD	101	User state change requested via remote diagnostic command bit	Não Configurado para outros estados, comando de troca de estado via bit remoto de diagnostico.
NCF_LCD_MENU_USER_CMD	102	User state change requested via LCD menu	Não Configurado para outros estados, comando de troca de estado via menu do LCD.
NCF_AUTO_CONFIGURATION	103	Auto configuration	Não Configurado para outros estados, auto configuração da redundância.
STR_STANDBY	200	Other CPU is controlling the system	Inicializando para Reserva, a outra UCP está controlando o sistema.
STR_ACTIVE	201	CPU started controlling the system	Inicializando para Ativo, UCP iniciou controlando o sistema em Ativo.
STR_CONFIG_ERROR	202	The Application has a configuration error	Inicializando para Não Configurado, a aplicação possui erro de configuração.
STR_LINK_WITH_INTERNAL_FAILURE	203	Redundant link with an internal failure	Inicializando para Inativo, canal de redundância interna com falha.
STR_NO_COMM_OTHER_CPU_AND_BUS_IDLE	204	This CPU is unable to communicate with the other CPU via redundant link and the BUS is idle	Inicializando para Ativo, esta UCP não consegue se comunicar com a outra UCP por canal de redundância e o barramento esta parado.
STR_NO_COMM_OTHER_CPU_AND_BUS_BUSY	205	This CPU is unable to communicate with the other CPU via redundant link and the BUS is busy	Inicializando para Inativo, esta UCP não consegue se comunicar com a outra UCP por canal de redundância e o barramento esta em uso.
STR_INCOMPATIBLE_FIRMWARE	206	The firmware version of this CPU is not compatible with the other one	Inicializando para Inativo, a versão de firmware desta UCP não é compatível com a da outra UCP.
STR_APPLICATION_DIFF	207	The Application between the CPUs is different	Inicializando para Não Configurado, a aplicação entre as UCPs está diferente.
STR_ARCHIVE_DIFF	208	The Project Archive between the CPUs is different	Inicializando para Não Configurado, o Project Archive entre as UCPs está diferente.
STR_ONLINE_CHANGE_APPLY	209	There is an online change in the Application of the Active CPU	Inicializando para Não configurado, uma carga online de aplicação foi feita na UCP em Ativo.
STR_DATA_EXCHANGE_FAILED	210	Redundant data synchronization failed	Inicializando para Inativo, sincronismo de dados redundantes falhou.
STR_APP_STATE_DIFF_WITH_OTHER_ACT	211	The application state of this CPU is different from the other Active CPU	Inicializando para Não Configurado, o estado da aplicação desta UCP é diferente da outra UCP em ativo.
STR_APP_STATE_DIFF_WITH_OTHER_NON_ACT	212	The application state of this CPU is different from the other Non-Active CPU	Inicializando para Ativo, o estado da aplicação desta UCP é diferente da outra UCP não ativa.
STR_BUS_ERROR	213	Bus error detected	Inicializando para Inativo, erro de barramento detectado.
STR_OTHER_CPU_ACT_WITH_BRKP	214	Other CPU is Active and with Application stopped in a Breakpoint	Inicializando para Inativo, A outra UCP está em Ativo e com a aplicação parada em Breakpoint.
SBY_LINK_WITH_INTERNAL_FAILURE	301	Redundant link with an internal failure	Reserva para Ativo, canal de redundância com falha interma.
SBY_APPLICATION_DIFF	302	The Application between the CPUs is different	Reserva para Não Configurado, a aplicação entre as UCPs está diferente.
SBY_ARCHIVE_DIFF	303	The Project Archive between the CPUs is different	Reserva para Não Configurado, o Project Archive entre as UCPs está diferente.
SBY_ONLINE_CHANGE_APPLY	304	There is an online change in the Application of the Active CPU	Reserva para Não Configurado uma carga online de aplicação foi feita na UCP em Ativo.
SBY_DATA_EXCHANGE_FAILED	305	Redundant data synchronization failed	Reserva para Inativo, sincronismo de dados redundantes falhou.
SBY_VITAL_FAILURE	306	The other CPU wasn't Active	Reserva para Inativo, a outra UCP não estava Ativa.

SBY_OTHER_PLC_NON_ACTIVE	307		Reserva para Ativo, a outra UCP não estava Ativa.
SBY_DIAGNOSTICS_FAILED	308	Exchange of Redundancy diagnostics failed	Reserva para Inativo, a outra UCP não estava Ativa.
SBY_BUS_ERROR	309	Bus error detected	Reserva para Inativo, erro de barramento detectado a outra UCP está ativa.
SBY_BUS_IDLE	310	BUS in idle state	Reserva para Ativo, a outra UCP não estava Ativa.
SBY_APP_STATE_DIFF_WITH_OTHER_ACT	311	The application state of this CPU is different from the other Active CPU	Reserva para Inativo, o estado da aplicação desta UCP é diferente da outra UCP em ativo.
SBY_LOCAL_CPU_USER_CMD	312	User state change requested via local diagnostic command bit	Outros estados para Reserva, Comando de troca de estado via bit local de diagnostico.
SBY_REMOTE_CPU_USER_CMD	313	User state change requested via remote diagnostic command bit	Outros estados para Reserva, Comando de troca de estado via bit remoto de diagnostico.
SBY_LCD_MENU_USER_CMD	314	User state change requested via LCD menu	Outros estados para Reserva, Comando de troca de estado via menu do LCD.
SBY_OTHER_CPU_ACT_WITH_BRKP	315	Other CPU is Active and with Application stopped in a Breakpoint	Reserva, A outra UCP está em Ativo e com a aplicação parada em Breakpoint.
INA_LOCAL_CPU_USER_CMD	401	User state change requested via local diagnostic command bit	Outros estados para Inativo, comando de troca de estado via bit local de diagnostico.
INA_REMOTE_CPU_USER_CMD	402	User state change requested via remote diagnostic command bit	Outros estados para Inativo, comando de troca de estado via bit remoto de diagnostico.
INA_LCD_MENU_USER_CMD	403	User state change requested via LCD menu	Outros estados para Inativo, Comando de troca de estado via menu do LCD.
ACT_LINK_WITH_INTERNAL_FAILURE	501	Redundancy internal link failure for one link	Houve erro de um canal interno de redundância.
ACT_BOTH_LINKS_WITH_INTERNAL_FAILURE	502	Redundancy internal link failure for both links	Houve erro em ambos os canais internos de redundância.
ACT_CONFLICT_ACTIVE	503	Both CPUs are in active redundancy state	Ativo para Reserva, ambas as UCPs estão em ativo.
ACT_VITAL_FAILURE	504	Both CPUs are in active redundancy state and Application state is equal	Ativo para Inativo, ambas as UCPs estão em ativo e com a aplicação no mesmo estado.
ACT_CONFLICT_ACTIVE_APP_STATE_EQUAL	505		Ativo para Reserva, ambas as UCPs estão em ativo e com a aplicação no mesmo estado.
ACT_CONFLICT_ACTIVE_APP_STATE_DIFF	506	Both CPUs are in active redundancy state and Application state is different	
ACT_BUS_ERROR	507	Bus error detected	Ativo para Inativo, erro de barramento detectado a outra UCP está em Reserva.
ACT_LOCAL_CPU_USER_CMD	508	User state change requested via local diagnostic command bit	Outros estados para Ativo, comando de troca de estado via bit local de diagnostico.
ACT_REMOTE_CPU_USER_CMD	509	User state change requested via remote diagnostic command bit	Outros estados para Ativo, comando de troca de estado via bit remoto de diagnostico.
ACT_LCD_MENU_USER_CMD	510	User state change requested via LCD menu	Outros estados para Ativo, Comando de troca de estado via menu do LCD.
ACT_APP_DELETE	511	Application of this CPU was deleted	A aplicação desta UCP foi apagada.
ACT_APP_RESET	512	Application of this CPU was reset	A Aplicação desta UCP sofreu reset.
ACT_APP_EXCEPTION	513	Application of this CPU entered in exception state	A aplicação desta UCP entrou em estado de exceção.
NON_ACT_OTHER_CPU_APP_DELETED_BY_USER	600	Application of the other CPU was deleted	A aplicação da outra UCP foi apagada.
NON_ACT_OTHER_CPU_APP_RESET_BY_USER	601	Application of the other CPU was reset	A Aplicação da outra UCP sofreu reset.
NON_ACT_OTHER_CPU_APP_BRKP_BY_USER	602	Application of the other CPU is stopped by breakpoint	Aplicação da outra UCP está em breakpoint.

Tabela 6-7. Razões de Troca de Estados da Redundância

Comandos da Redundância

Os campos de comandos das estruturas RedCmdLoc e RedCmdRem possuem sempre um sufixo que pode ser Local ou Remote. Por exemplo, existem os campos de comando StandbyLocal e StandbyRemote, que têm efeito equivalente ao comando de "Trocar para Reserva" do visor da UCP.

Um comando com sufixo Local gerado em RedCmdLoc será executado na própria UCP (local).

Por outro lado, um comando com sufixo Remote gerado em RedCmdLoc será executado na outra UCP (remota). Isto funciona da seguinte maneira:

- A UCP remota, a cada ciclo da MainTask, recebe uma cópia de RedCmdLoc da UCP local via canais de sincronismo, e esta cópia é chamada de RedCmdRem na UCP remota
- A UCP remota somente executa comandos de RedCmdRem que tenham o sufixo Remote

Exemplo 1: se a UCP local estiver em estado Ativo, e deseja-se chaveá-la para o estado Reserva, deve-se ligar o bit DG_HX3040_01.RedCmdLoc.bStandbyLocal na UCP local.

Exemplo 2: se a UCP remota estiver em estado Ativo, e deseja-se chaveá-la para o estado Reserva, deve-se ligar o bit DG_HX3040_01.RedCmdLoc.bStandbyRemote na UCP local.

ATENÇÃO:
Se o diagnóstico DG_HX3040_01.RedDgnLoc.RedundancyDiagnostics.bExchangeSync estiver indicando falha no serviço Troca de Diagnósticos e Comandos, um comando com sufixo Remote não poderá ser repassado para a UCP remota, e portanto não será executado.

Para disparar um comando, deve-se sempre ligar o bit correspondente em RedCmdLoc. Isto pode ser feito por um sistema SCADA, fazendo uma escrita via MasterTool, ou até mesmo ligando o bit dentro de uma POU como UserPrg ou NonSkippedPrg.

O usuário não precisa se preocupar com o desligamento do bit de comando, que será feito automaticamente pelo gerenciador de redundância:

- No caso de comandos executados na UCP local (RedCmdLoc + comando com sufixo Local), o bit é desligado assim que o comando for percebido e executado
- No caso de comandos executados na UCP remota (RedCmdRem + comando com sufixo Remote):
    - Na UCP remota, o comando é executado quando o gerenciador de redundância percebe uma borda de subida no bit de comando

A seguir, são definidos os campos das estruturas RedCmdLoc e RedCmdRem:

.RedCmdLoc.* .RedCmdRem.*	Tipo	Descrição
bAutoConfigLocal	BOOL	TRUE – Este diagnóstico informa uma solicitação de configuração automática, necessária para deixar o estado Não-Configurado em algumas situações.
		FALSE – Solicitação de configuração automática desabilitada.
bStandbyLocal	BOOL	TRUE – Este comando produz uma ação equivalente à solicitação de "Trocar para Reserva" do menu Redundancia, disponível no visor da UCP.
		FALSE – Não há solicitação pendente.
bMenuStandbyLocal	BOOL	TRUE – Liga por um ciclo da MainTask, indicando que o usuário solicitou um comando de "Trocar para Reserva" através do Menu do Visor da UCP.
		FALSE – Não há solicitação pendente.
bInactiveLocal	BOOL	TRUE – Este comando produz uma ação equivalente à solicitação de "Trocar para Inativo" do menu Redundancia, disponível no visor da UCP.
		FALSE – Não há solicitação pendente.
bMenuInactiveLocal	BOOL	TRUE – Liga por um ciclo da MainTask, indicando que o usuário solicitou um comando de "Trocar para Inativo" através do Menu do Visor da UCP.
		FALSE – Não há solicitação pendente.
bStandbyRemote	BOOL	TRUE – Este comando produz uma ação equivalente à solicitação de "Trocar para Reserva" do menu Redundancia, disponível no visor da UCP Remota.
		FALSE – Não há solicitação pendente.

bInactiveRemote	BOOL	TRUE – Este comando produz uma ação equivalente à solicitação de “Trocar para Inativo” do menu Redundancia, disponível no visor da UCP.remota.
		FALSE – Não há solicitação pendente.
bResetNETStatisticsLocal	BOOL	TRUE – Este comando reseta as estatísticas dos canais de sincronismo de redundância A/B (veja subestrutura SyncLinkStatistics em RedDgnLoc e RedDgnRem). Tais estatísticas são contadores de falhas e sucessos em serviços de sincronização.
		FALSE – O comando de reset das estatísticas dos canais de sincronismo A/B na UCP local não foi acionado.
bResetNETStatisticsRemote	BOOL	TRUE – Produz uma ação equivalente ao comando ResetNETStatisticsLocal, porém desta vez na UCP remota.
		FALSE – O comando de reset das estatísticas dos canais de sincronismo A/B na UCP remota não foi acionado.

Tabela 6-8. Comandos da Redundância

Informações do Usuário Trocados entre UCPA e UCPB

O serviço de sincronização Troca de Diagnósticos e Comandos, troca as seguintes estruturas de dados entre as duas UCPs em cada ciclo da MainTask, usando os canais de sincronismo A/B:

- Diagnósticos de Redundância (RedDgnLoc e RedDgnRem), já discutidos na seção Estrutura de Diagnósticos da Redundância
- Comandos de Redundância (RedCmdLoc e RedCmdRem), já discutidos na seção Comandos da Redundância
- Informações do Usuário Trocados entre UCPA e UCPB (RedUsrLoc e RedUsrRem), que serão discutidos nesta seção

As estruturas RedUsrLoc e RedUsrRem são simplesmente um array de 128 bytes, cuja utilização pode ser livremente definida pelo usuário. Elas permitem que o usuário transfira, a cada ciclo, 128 bytes de informação da UCPA para a UCPB, e outros 128 bytes da UCPB para a UCPA.

RedUsrRem é uma cópia de RedUsrLoc da outra UCP, recebida via canais de sincronismo internos. Determinada UCP escreve informações em RedUsrLoc, que serão recebidas na outra UCP em RedUsrRem.

Logs de Eventos da Redundância

O MasterTool permite observar diversos logs para uma UCP Hadron Xtorm, entre os quais encontra-se os logs de eventos da redundância. Estas mensagens, específicas para a redundância, registram no Log de Sistema.

São registradas modificações relevantes nos campos das estruturas de dados de diagnósticos e comandos de redundância, que são as seguintes:

- RedDgnLoc
- RedDgnRem
- RedCmdLoc
- RedCmdRem

No caso das estruturas de diagnósticos, apenas os seguintes campos não geram logs de eventos:

- wRedStateDuration
- wCycleCounter
- dwIECTimer
- SyncLinkStatistics

Cada linha mostrada no log possui as seguintes colunas:

- Marca de Tempo: data e hora do evento, com resolução de milissegundos
- Severidade: informação, advertência, erro ou exceção
- Descrição: texto que descreve o evento

- Componente: componente que gerou o evento, que no caso do log de eventos da redundância será o “Redundancy Management”

O texto na coluna “Descrição” contém informações referentes ao evento. Por exemplo, no caso da troca de estado da redundância de Reserva para Ativo, será gerada uma entrada no log indicando que houve a troca de estado, do estado Reserva para o Ativo.

Para acessar esta tela, deve-se dar um duplo clique sobre o dispositivo (HX3040) na árvore de dispositivos, e depois selecionar a aba “Log”. Existe um filtro que permite selecionar somente o componente “Redundancy Management”, para exibir somente os eventos de redundância.

ATENÇÃO:
Alguns diagnósticos podem apontar possíveis falhas durante a inicialização do sistema redundante e nos primeiro ciclos das tarefas. Mas, em um correto funcionamento do sistema, estes diagnósticos voltam a indicar a ausência de erros logo após a inicialização do sistema.

7. Manutenção

ATENÇÃO:
Para uma manutenção adequada e com segurança, deve-se consultar primeiramente o subcapítulo de Segurança Elétrica, parte integrante do capítulo Instalação.

Diagnósticos do Módulo UCP

Uma das características da Série Hadron Xtorm é a geração de diagnósticos de comportamentos atípicos, sejam eles falhas, erros ou modos de operação, possibilitando ao operador identificar e solucionar problemas que venham a ocorrer com o sistema com grande facilidade.

A UCP Hadron Xtorm proporcionam diversas maneiras de se visualizar os diagnósticos gerados pelo sistema, são elas:

- One Touch Diag
- Diagnósticos via LED
- Diagnósticos via WEB
- Diagnósticos via Variáveis
- Diagnósticos via Blocos Funcionais

O primeiro deles é uma característica da Série Hadron Xtorm, a qual possibilita um rápido acesso às condições anormais da aplicação. O segundo é puramente visual, gerado através de um LED bicolor (DL) (azul / vermelho) presente no frontal do módulo. A próxima característica é a visualização gráfica em uma página WEB do bastidor e os respectivos módulos configurados, sendo permitido o acesso individual do estado de operação e os diagnósticos ativos. Os diagnósticos também são armazenados diretamente em variáveis simbólicas da UCP, e podem ser utilizados pela aplicação do usuário, por exemplo, sendo apresentados em um sistema de supervisão. Os últimos apresentam condições específicas de funcionamento do sistema.

A função destes diagnósticos é apontar possíveis problemas de instalação ou configuração do sistema, e de problemas ou deficiências das redes de comunicação. O capítulo de Manutenção deve ser consultado pelo usuário sempre que necessário.

One Touch Diag

One Touch Diag (OTD), ou seja, diagnóstico com um único pressionamento, é uma característica que a Série Hadron Xtorm traz para a Unidade Terminal Remota. Com este conceito, o usuário pode verificar os diagnósticos de qualquer módulo presente no sistema diretamente no visor gráfico da UCP, com um único pressionamento no botão de diagnóstico do respectivo módulo. Essa é uma poderosa ferramenta de diagnóstico que pode ser usada off-line (sem sistema de supervisão ou software programador), tornando mais fácil encontrar e resolver rapidamente possíveis problemas.

O botão de diagnóstico está localizado na parte frontal da UCP, em local de fácil acesso, e, além de fornecer os diagnósticos ativos, permite o acesso ao menu de navegação, descrito no capítulo Menu Informativo e de Configuração da UCP.

A Figura 7-1 mostra a localização da tecla na UCP:

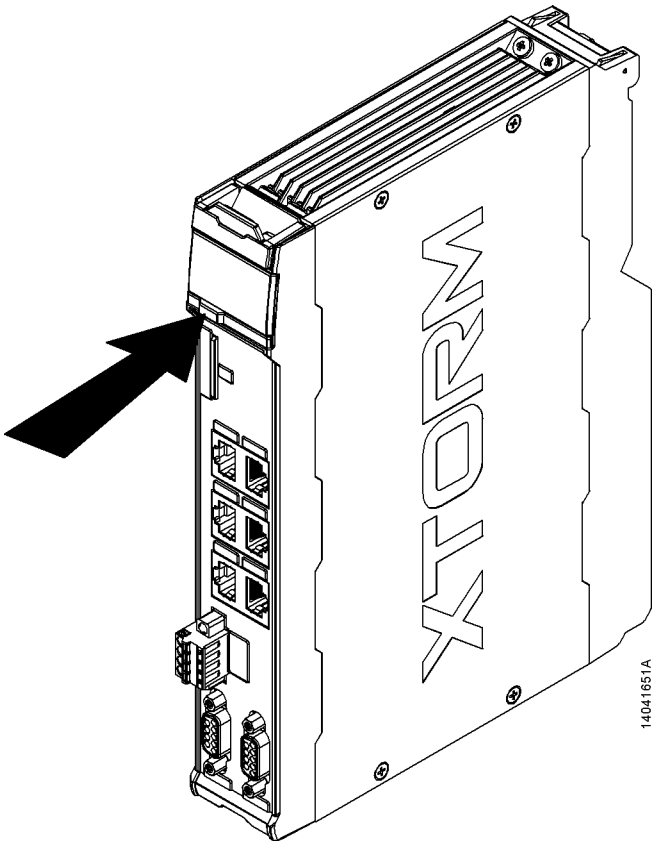


Figura 7-1. Botão de Diagnóstico

Com apenas um pressionamento curto, a UCP começa a mostrar os diagnósticos do barramento (quando ativos, caso contrário exibe a mensagem “SEM DIAG”). Inicialmente, será visualizada a Tag (configurada nas propriedades do módulo no software MasterTool Xtorm, seguindo a IEC 61131-3), ou seja, o nome atribuído à UCP, em seguida, serão mostrados todos os diagnósticos, através de mensagens no visor da UCP. Esse processo será executado por duas vezes no visor. Tudo ocorre de forma automática, sendo que o usuário somente deverá executar o pressionamento curto inicial e a UCP será responsável por exibir os diagnósticos. Os diagnósticos de outros módulos presentes no barramento também serão exibidos no visor gráfico da UCP, através de um pressionamento curto no botão de diagnóstico dos mesmos, no mesmo modelo da apresentação dos diagnósticos da UCP.

A Figura 7-2 mostra todo o processo a partir do pressionamento curto, sendo a condição e os tempos da UCP representados nos retângulos menores. É importante salientar que os diagnósticos poderão ter mais de uma tela, ou seja, o tempo especificado no fluxograma abaixo é válido para cada uma delas.

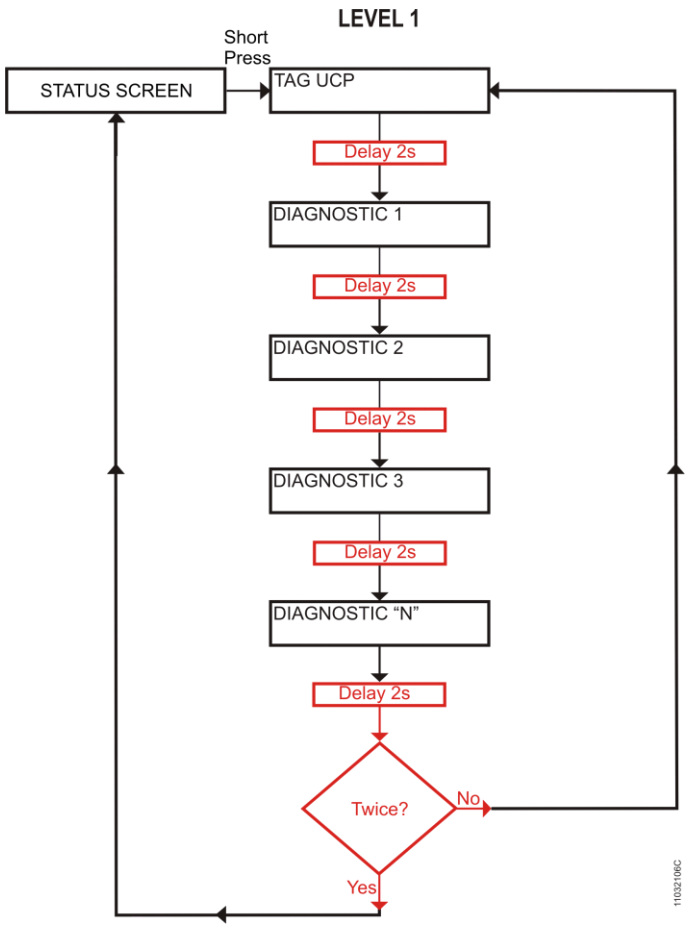


Figura 7-2. Visualização dos Diagnósticos da UCP

Para finalizar, antes de todo o processo de visualização ser efetuado, basta dar um pressionamento curto no botão de diagnóstico, em qualquer momento, ou pressionar o botão diagnóstico de algum módulo de E/S presente no barramento.

Caso seja efetuado um pressionamento longo, a UCP entrará no menu de navegação, o qual está descrito no capítulo Menu Informativo e de Configuração da UCP.

A Tabela 7-1 mostra a diferença entre os tempos do pressionamento curto, pressionamento longo e botão preso.

Tipo de Pressionamento	Tempo Mínimo	Tempo Máximo	Condição para Indicação
Sem pressionamento	-	59,99 ms	-
Pressionamento curto	60 ms	0,99 s	Liberação
Pressionamento longo	1 s	20 s	Mais do que 1 s até 20 s
Tecla Presa	20,01 s	(∞)	Indicação em diagnóstico, consultar a Tabela 7-4

Tabela 7-1. Tempo de One Touch

As mensagens exibidas no visor gráfico da UCP Hadron Xtorm, correspondentes aos diagnósticos, estão descritas na seção Diagnósticos via Variáveis, na Tabela 7-3 e Tabela 7-4.

Caso ocorra uma situação de tecla presa de algum módulo de E/S presente no barramento, o botão de diagnóstico deste deixará de indicar diagnósticos no visor gráfico da UCP quando for pressionado, neste caso a UCP irá indicar que há módulos com diagnóstico ativo. Para que seja possível eliminar

este diagnóstico da UCP, será necessário realizar uma troca a quente do módulo com o diagnóstico ativo.

Notas:

One Touch Diag (OTD): Essa opção somente estará disponível ao usuário quando o módulo estiver em modo operacional.

Diagnósticos via LED

A UCP da Série Hadron Xtorm possui um LED (DL) para indicação de diagnósticos. A Tabela 7-2 mostra o significado de cada estado do LED.

DL (Cor)	Descrição	Causas	Prioridade
Desligado	Módulo desligado ou falha no visor	Sem fonte de alimentação Problema de hardware	-
Ligado (Azul)	Todas as aplicações em execução (Modo Run)	-	4 (Mais Baixo)
Piscando 2x (Azul)	Módulos do barramento ou UCP com diagnóstico	No mínimo um módulo do barramento, incluindo a UCP, está com algum diagnóstico ativo	2
Piscando 3x (Azul)	Forçamento de dados	Alguma área de memória está sendo forçada pelo usuário via MasterTool Xtorm	3
Ligado (Vermelho)	Todas as aplicações paradas (Modo Stop)	-	4 (Mais Baixo)
Piscando 1x (Vermelho)	Cão-de-guarda de Software	Cão-de-guarda da aplicação de usuário	1
Piscando 4x (Vermelho)	Erro de Configuração ou de Hardware no barramento.	O barramento está danificado ou não está configurado corretamente	0 (Mais Alta)

Tabela 7-2. Descrição dos Estados do LED de Diagnóstico

Notas

Cão-de-guarda de software: Para remover a indicação de cão-de-guarda, deve-se efetuar um reset da aplicação ou desligar e ligar novamente a UCP. Esse cão-de-guarda ocorre quando o tempo de execução da aplicação de usuário for maior que o tempo de cão-de-guarda configurado.

Diagnósticos via WEB

Além das características apresentadas anteriormente, a Série Hadron Xtorm traz para o usuário uma ferramenta inovadora de acesso aos diagnósticos e estados de operação do sistema, através de uma página WEB.

A utilização, além de dinâmica, é bastante intuitiva e facilita as operações do usuário. Em outras palavras, pode substituir o uso de um sistema de supervisão quando o uso for restrito a verificação de status do sistema.

Para acessar a página WEB da UCP desejada, basta utilizar um navegador padrão (Internet Explorer, Mozilla Firefox 3.0 ou superior e Google Chrome 8 ou superior) e digitar, na barra de endereço, o endereço IP correspondente a UCP (Ex.: http://192.168.1.1). Inicialmente, será apresentada as informações da UCP, conforme mostra a Figura 7-3:

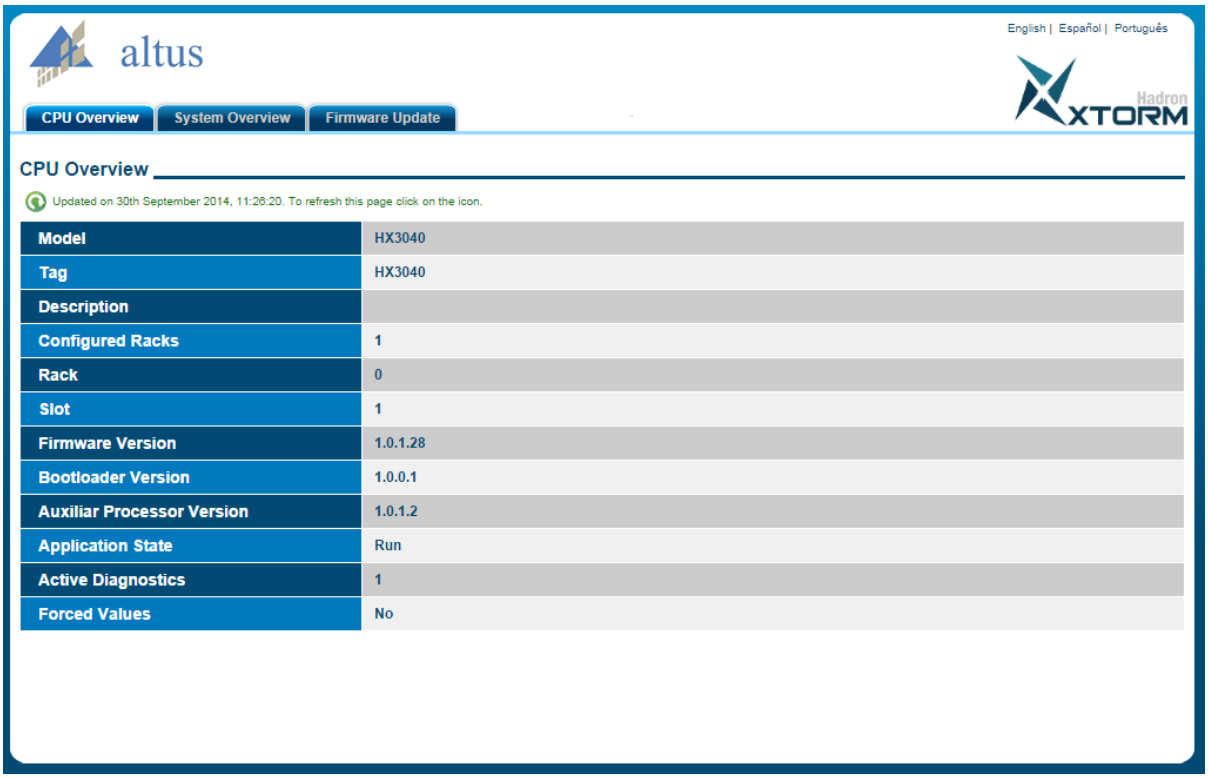


Figura 7-3. Tela Inicial

Também existe a aba “System Overview”, a qual pode ser visualizada através do Bastidor ou da lista dos módulos presentes (opção do lado direito da tela). Quando não houver nenhuma aplicação na UCP, será exibida nesta página uma configuração com o maior Bastidor disponível e uma fonte de alimentação padrão, juntamente com a UCP conectada. Quando a visualização pelo bastidor é utilizada, os módulos que têm diagnóstico ficam piscando e assumem a cor vermelha, conforme mostra a Figura 7-4. Caso contrário será exibida uma lista com os módulos presentes no sistema, Tags correspondentes e número de diagnósticos ativos:

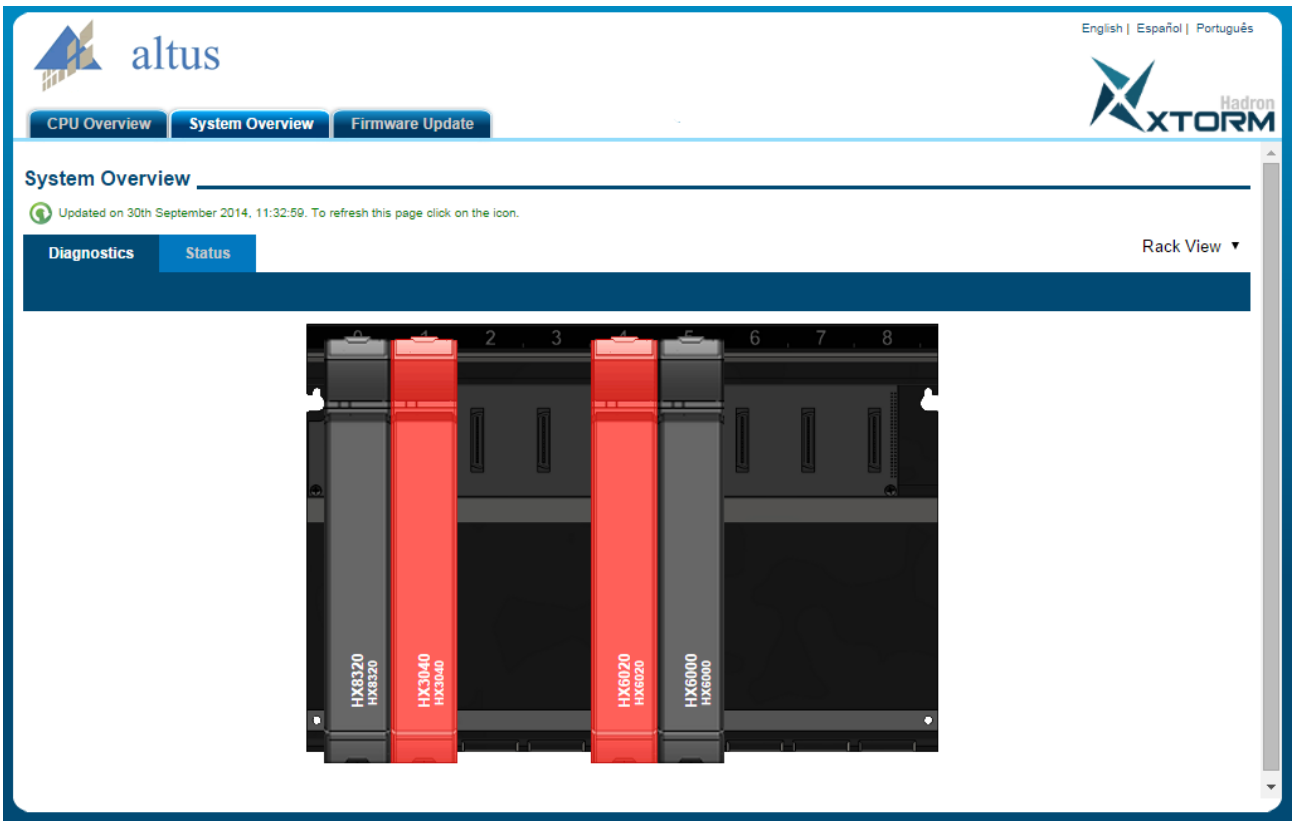


Figura 7-4. Informações do Sistema

Ao clicar em um dos módulos, no mesmo instante é mostrada uma tela que mostra a quantidade de diagnósticos do módulo e a quantidade de diagnóstico para cada canal do módulo, conforme mostra a Figura 7-5:

ATENÇÃO: Quando uma UCP for reiniciada e a aplicação entrar em exceção na partida do sistema, os diagnósticos não estarão válidos. É necessário corrigir o problema que gera a exceção da aplicação para que os diagnósticos sejam atualizados.

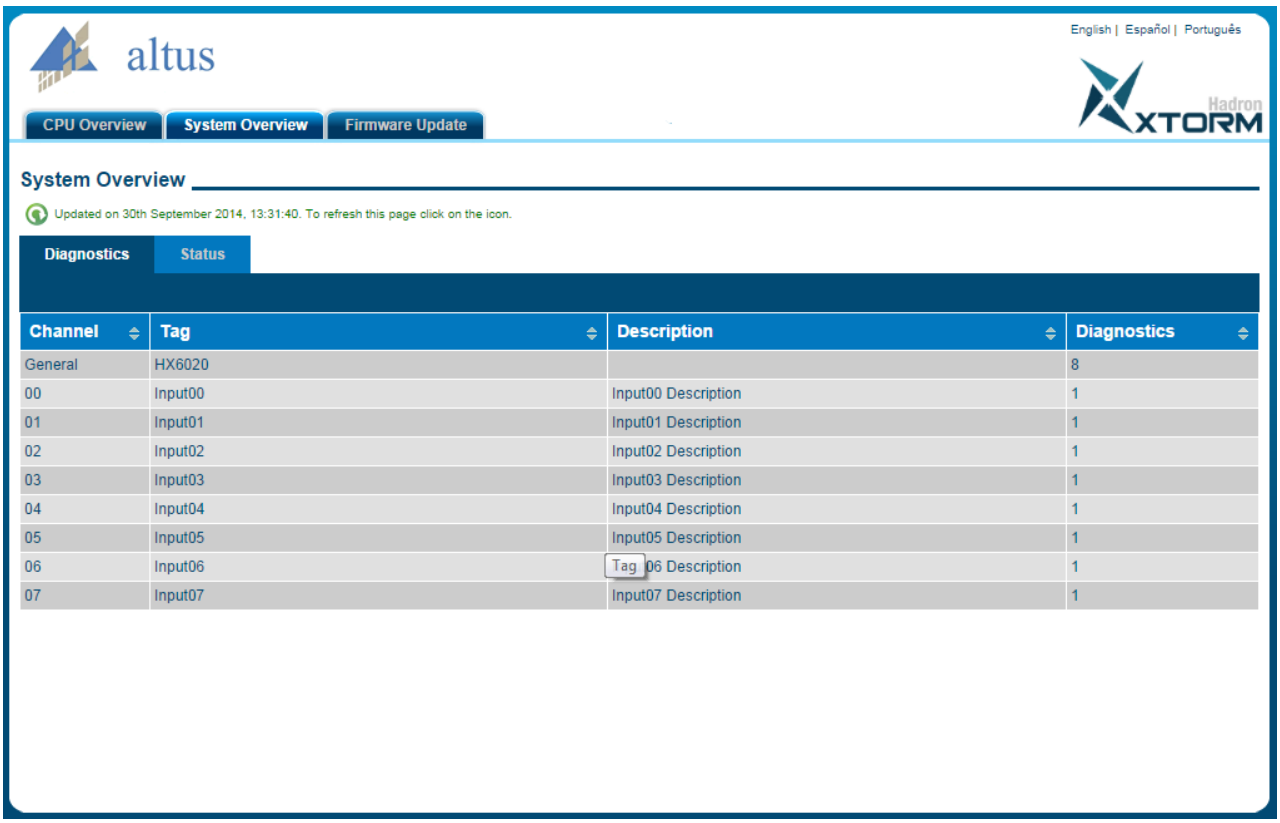


Figura 7-5. Quantidade de Diagnósticos por módulo e por canal

Ao clicar em uma das linhas da tabela da Figura 7-5, no mesmo instante é mostrada uma tela que mostra a descrição de cada diagnóstico, conforme mostra a Figura 7-6:

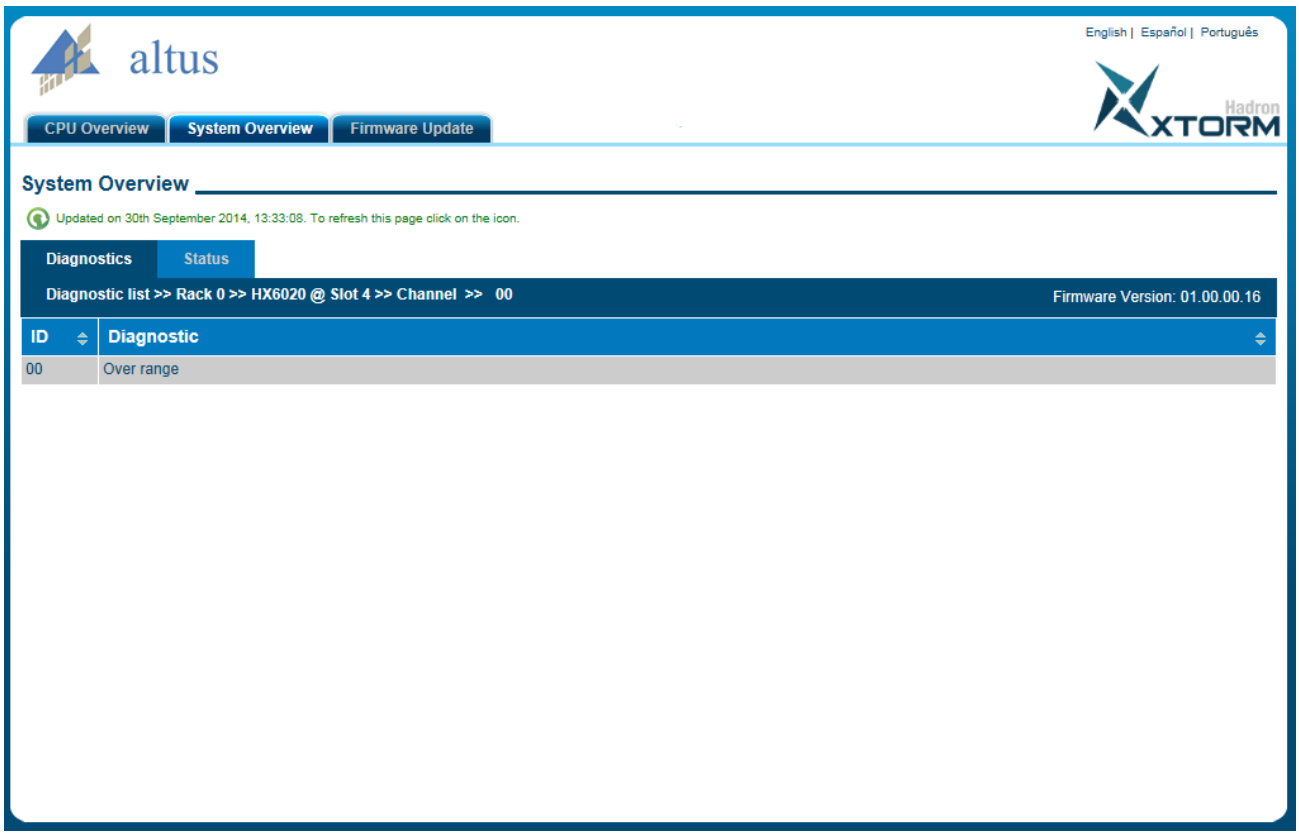


Figura 7-6. Lista de Diagnósticos do Módulo/Canal

ATENÇÃO:
A lista completa de diagnósticos de cada módulo da Série Xtorm pode ser encontrado no documento de Características Técnicas de cada módulo, cujos documentos estão listados na Tabela 1-1. Documentos Relacionados.

Caso a aba Status seja selecionada, o estado de todos os diagnósticos detalhados é exibido na tela.

Além disso, o usuário pode optar por três opções de idioma: Português, Inglês e Espanhol. Basta alterar o menu superior direito para o idioma desejado.

A aba Atualização de Firmware é restrita ao usuário, ou seja, apenas para uso interno da Altus. Em casos onde a atualização é realizada remotamente (através de uma conexão de rádio ou satélite por exemplo, a velocidade mínima deste link deve ser de 128Kbps).

ATENÇÃO:
A visualização dos diagnósticos Web é compatível somente até a versão 9 do navegador Internet Explorer, sendo que nesta versão é necessário habilitar o modo de exibição de compatibilidade do navegador. Caso contrário à exibição de algumas páginas poderá ser afetada.

Diagnostic Explorer

O Diagnostic Explorer é a inclusão dos diagnósticos via WEB dentro do MasterTool Xtorm, a fim de que o acesso seja mais rápido e objetivo.

O acesso à característica ocorre de duas maneiras:

- Acessando a opção “Diagnostic Explorer” na árvore de dispositivos, localizada na esquerda da tela do MasterTool Xtorm, colocando o IP correto no campo indicado na Figura 7-7. Lembrando que para a página de diagnósticos ser exibida, o usuário deverá estar conectado à UCP (capítulo Programação Inicial - Login).

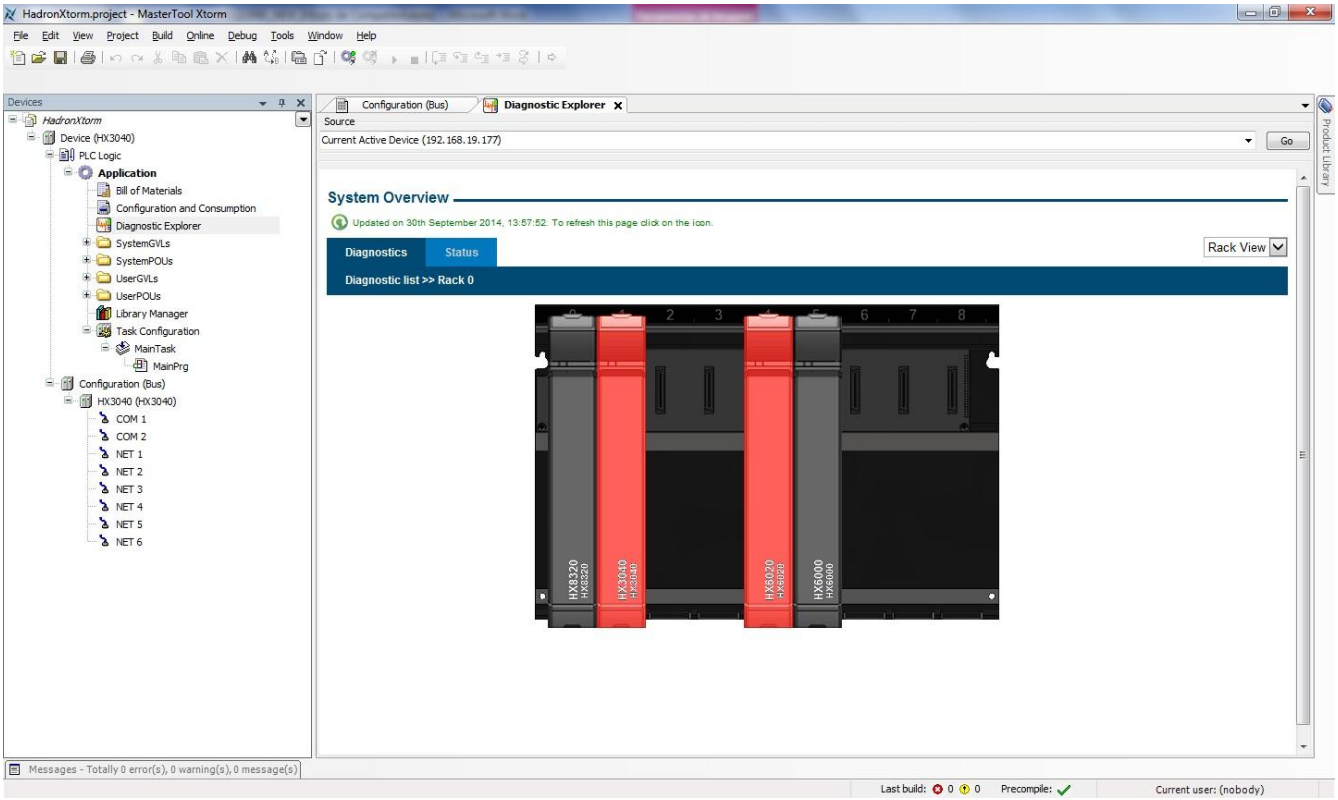


Figura 7-7. Tela do Diagnostic Explorer

- Clicando com o botão direito do mouse sobre o módulo, e selecionando “Diagnósticos”, o Diagnostic Explorer será aberto, direcionando para a página de status do respectivo módulo

Diagnósticos via Variáveis

A UCP da Série Hadron Xtorm possui um conjunto de variáveis simbólicas globais onde são disponibilizadas uma série de informações de diagnóstico relacionadas ao hardware e software. Estas estruturas de dados são para indicação de diagnóstico de todos os módulos declarados no barramento, mapeadas sobre variáveis de representação direta %Q, e definidas simbolicamente através da diretiva AT, na GLV Diagnostics criada automaticamente pelo MasterTool IEC XE.

Diagnósticos Resumidos

A Tabela 7-3 mostra o significado de cada das variáveis de diagnósticos resumidos da UCP:

Mensagem de Diagnóstico	DG_modulename.tSummarized.*	Tipo	Descrição
SEM DIAG	-		Não existe diagnóstico ativo.
CONFIG. INCOMPATIVEL	bConfigMismatch	BOOL	TRUE – Existe algum problema de configuração no barramento, como módulo em posição incorreta.
			FALSE – O barramento está configurado corretamente.
MODULOS AUSENTES	bAbsentModules	BOOL	TRUE – Um ou mais módulos declarados estão ausentes.
			FALSE – Todos os módulos estão presentes no barramento.
MODULOS TROCADOS	bSwappedModules	BOOL	TRUE – Dois módulos estão trocados entre si no barramento.
			FALSE – Não há módulos trocados no barramento.
MODULOS NÃO DECLARADOS	bNonDeclaredModules	BOOL	TRUE – Um ou mais módulos presentes no barramento não estão declarados
			FALSE – Todos os módulos presentes no barramento estão declarados
MODULOS C/ DIAGNOSTICO	bModulesWithDiagnostic	BOOL	TRUE – Um ou mais módulos do barramento estão com diagnóstico ativo.
			FALSE – Não existem diagnósticos ativos nos módulos do barramento.
MODULOS C/ ERRO FATAL	bModuleFatalError	BOOL	TRUE – Um ou mais módulos presentes no barramento estão em estado não funcional.
			FALSE – Todos os módulos presentes no barramento estão em estado funcional.
MODULOS C/ ERRO PARAM.	bModuleParameterError	BOOL	TRUE – Um ou mais módulos do barramento estão com erro de parametrização.
			FALSE – Todos os módulos estão parametrizados.
ERRO NO BARRAMENTO	bWHSBBusError	BOOL	TRUE – Indicação do mestre que existe falha no barramento WHSB.
			FALSE – O barramento WHSB está em correto funcionamento.
BASTIDOR AUSENTE	bAbsentRacks	BOOL	TRUE – Um ou mais bastidores declarados estão ausentes.
			FALSE – Todos os bastidores declarados estão presentes.
BASTIDOR DUPLICADO	bDuplicatedRacks	BOOL	TRUE – Há algum bastidor com o número de identificação duplicado.
			FALSE – Não existem bastidores com o número de identificação duplicado.
BASTIDOR INVALIDO	bInvalidRacks	BOOL	TRUE – Há algum bastidor com o número de identificação inválido.
			FALSE – Não existem bastidores com o número de identificação inválido.
BASTIDOR NÃO DECLARADO	bNonDeclaredRacks	BOOL	TRUE – Há algum bastidor com o número de identificação não declarado.
			FALSE – Não existem bastidores com o número de identificação não declarado.
SLOT DUPLICADO	bDuplicatedSlots	BOOL	TRUE – Há algum endereço de posição duplicado
			FALSE – Não existem endereços de posição duplicados
-	bReserved_13..15	BOOL	Reservado
FALHA DE HARDWARE	bHardwareFailure	BOOL	TRUE – Falha no hardware da UCP.
			FALSE – O hardware está em correto funcionamento.
EXCECAO NO SOFTWARE	bSoftwareException	BOOL	TRUE – Uma ou mais exceções geradas pelo software.
			FALSE – Não foram geradas exceções no software.

-	bReserved_18	BOOL	Reservado
ERRO CARTAO DE MEMORIA	bMemoryCardError	BOOL	TRUE – O cartão de memória está inserido na UCP, porém não está funcionando corretamente.
			FALSE – O cartão de memória está funcionando corretamente.
-	bReserved_20..23	BOOL	Reservado
ERRO CONFIG. COM 1	bCOM1ConfigError	BOOL	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface serial COM 1.
			FALSE – A configuração da interface serial COM 1 está correta.
ERRO CONFIG. COM 2	bCOM2ConfigError	BOOL	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface serial COM 2.
			FALSE – A configuração da interface serial COM 2 está correta.
ERRO CONFIG. NET 1	bNET1ConfigError	BOOL	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 1.
			FALSE – A configuração da interface Ethernet NET 1 está correta.
ERRO CONFIG. NET 2	bNET2ConfigError	BOOL	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 2.
			FALSE – A configuração da interface Ethernet NET 2 está correta.
ERRO CONFIG. NET 3	bNET3ConfigError	BOOL	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 3.
			FALSE – A configuração da interface Ethernet NET 3 está correta.
ERRO CONFIG. NET 4	bNET4ConfigError	BOOL	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 4.
			FALSE – A configuração da interface Ethernet NET 4 está correta.
ERRO CONFIG. NET 5	bNET5ConfigError	BOOL	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 5.
			FALSE – A configuração da interface Ethernet NET 5 está correta.
ERRO CONFIG. NET 6	bNET6ConfigError	BOOL	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 6.
			FALSE – A configuração da interface Ethernet NET 6 está correta.
DATA/HORA INVALIDA	bInvalidDateTime	BOOL	TRUE – A data ou a hora são inválidas.
			FALSE – A data e a hora estão corretas.
RUNTIME RESET	bRuntimeReset	BOOL	TRUE – O RTS (Runtime System) foi reiniciado pelo menos uma vez. Esse diagnóstico somente é limpo na reinicialização do sistema.
			FALSE – O RTS (Runtime System) está operando normalmente.
ERRO TECLA OTD	bOTDSwitchError	BOOL	TRUE –A tecla ficou travada por mais de 20 s pelo menos uma vez enquanto a UCP estava energizada. Esse diagnóstico somente é limpo na reinicialização do sistema.
PERDA DE RETENTIVIDADE	bRetentivityLost	BOOL	TRUE – Dados inválidos encontrados na memória retentiva durante a inicialização.
UCP POSICAO INCORRETA	bWrongCPUSlot	BOOL	TRUE – UCP posicionada em posição diferente da configurada via programador
			FALSE – UCP posicionada na posição correta, conforme configurada via programador
-	bReserved_37..39	BOOL	Reservado

Tabela 7-3. Tabela dos Diagnósticos Resumidos da UCP

Notas:

Sem Diag: Não existe diagnóstico ativo na UCP

Configuração incompatível: O diagnóstico de configuração incompatível é gerado caso um ou mais módulos presentes (fisicamente), não confere(m) com o que foi declarado. O diagnóstico de módulos trocados ligará junto caso haja a troca de 2 módulos entre si (é uma informação complementar a configuração incompatível). Os módulos inseridos no barramento, além dos módulos declarados no projeto, não ligam este bit de diagnóstico, apenas em caso de módulos ausentes ou diferentes.

Módulos Ausentes: Se um ou mais módulos declarados estão ausentes no barramento então o diagnóstico de Módulos Ausentes será verdadeiro.

Módulos trocados: Se somente dois módulos estiverem trocados entre si no respectivo barramento, então o diagnóstico de troca pode ser identificado. Caso contrário, o problema será tratado como "Configuração Incompatível".

Módulos não declarados: Se um ou mais módulos presentes no barramento não estão declarados então o diagnóstico de Módulos não declarados será verdadeiro.

Módulos com diagnósticos: Se um ou mais módulos presentes no barramento estão com diagnósticos ativos então o diagnóstico de Módulos com diagnósticos será verdadeiro.

Módulos c/ erro fatal: Caso o diagnóstico de Módulos com Erro Fatal seja verdadeiro, verificar qual é o módulo que está com problema no barramento e encaminhar para a Assistência Técnica da Altus, pois o mesmo está apresentando falha do hardware.

Módulos c/ erro param: Caso o diagnóstico de Módulo com Erro de Parametrização seja verdadeiro, verificar se os módulos do barramento estão configurados corretamente e se as versões de firmware e do software MasterTool Xtorm estão adequadas.

Erro no barramento: Considerado um erro fatal, interrompendo o acesso aos módulos do barramento. Caso o diagnóstico de erro no barramento seja verdadeiro, talvez uma situação anormal devido a configuração de troca a quente selecionada pode ter acontecido ou talvez haja um problema de hardware nas linhas de comunicação do barramento, sendo assim, deve-se entrar em contato com a Assistência Técnica da Altus.

Bastidor Ausente: Se um ou mais bastidores declarados estão ausentes então o diagnóstico de Bastidor Ausente será verdadeiro.

Bastidor Duplicado: Se há algum bastidor com o número de identificação duplicado então o diagnóstico de Bastidor Duplicado será verdadeiro.

Bastidor inválido: Se há algum bastidor com o número de identificação inválido então o diagnóstico de Bastidor inválido será verdadeiro.

Bastidor não declarado: Se há algum bastidor com o número de identificação não declarado então o diagnóstico de Bastidor não declarado será verdadeiro.

Slot duplicado: Se há algum endereço de posição duplicado então o diagnóstico de Slot duplicado será verdadeiro.

Falha de hardware: Caso o diagnóstico de Falha de Hardware seja verdadeiro, encaminhar a UCP para Assistência Técnica da Altus, pois a mesma apresenta problemas no RTC, processador auxiliar, ou outros recursos de hardware.

Exceção no software: Caso o diagnóstico de exceção no software seja verdadeiro, o usuário deverá verificar a sua aplicação para garantir que a mesma não esteja acessando indevidamente a memória. Se o problema persistir, o setor de Suporte da Altus deverá ser consultado. Os códigos de exceção no software estão descritos após a tabela de diagnósticos detalhados da UCP.

Perda de Retentividade: Este diagnóstico indica que houve perda dos dados não voláteis (variáveis retentivas, persistentes e fila de eventos). É ligado somente quando a perda for causada por problemas de hardware (falha no hardware, ou remoção à quente da fonte de alimentação ou da UCP). Comandos de reset cold e reset origin disparados através da ferramenta MasterTool Xtorm não causam o acionamento deste diagnóstico.

Diagnósticos Detalhados

A Tabela 7-4 mostra os diagnósticos detalhados da UCP:

Variável DG_HX3040.tDetailed.*		Tipo	Descrição
Target.*	dwCPUModel	DWORD	HX3040 = 0x3040
	abyCPUVersion	BYTE ARRAY(4)	Versão do firmware.
	abyBootloaderVersion	BYTE ARRAY(4)	Versão do bootloader
	abyAuxprocVersion	BYTE ARRAY(4)	Versão do processador auxiliar

Hardware.*	bAuxprocFailure	BOOL	Falha na comunicação entre o processador auxiliar e o processador principal.
	bRTCFailure	BOOL	O processador principal não está habilitado para comunicar com o RTC (relógio da UCP).
	bThermometerFailure	BOOL	Falha na comunicação entre o termômetro e o processador principal.
	bLCDFailure	BOOL	Falha na comunicação entre o visor LCD e o processador principal.
Exception.*	wExceptionCode	WORD	Código de Exceção gerado pelo RTS.
	byProcessorLoad	BYTE	Nível, em percentagem (%), de carga no processador.
RetainInfo.*	byCPUInitStatus	BYTE	Status da Inicialização da UCP: 01: Hot start 02: Warm Start 03: Cold Start Obs.: Essas variáveis são reinicializadas em todas as energizações.
	wCPUColdStartCounter	WORD	Contador de Inicializações à frio: Será incrementado somente devido a retirada a quente da UCP no barramento e não devido ao comando de Reset a Frio do MasterTool Xtorm. (0 a 65535)
	wCPUWarmStartCounter	WORD	Contador de Inicializações à quente: Será incrementado somente durante a sequência de energizações do sistema e não devido ao comando de Reset a Quente do MasterTool Xtorm. (0 a 65535)
	wCPUHotStartCounter	WORD	Contador de distúrbios menores do que o tempo de suporte a falhas na alimentação da UCP. (0 a 65535)
	wRTSResetCounter	WORD	Contador de reset efetuados pelo RTS (Runtime System). (0 a 65535)
Reset.*	bBrownOutReset	BOOL	A UCP foi reiniciada devido a uma falha na alimentação na última inicialização.
	bWatchdogReset	BOOL	A UCP foi reiniciada devido ao cão-de-guarda ativo na última inicialização.
Thermometer. *	bOverTemperatureAlarm	BOOL	Alarme gerado devido à temperatura interna estar em 85 ºC ou acima de 85 ºC.
	bUnderTemperatureAlarm	BOOL	Alarme gerado devido à temperatura interna estar em 0 ºC ou abaixo de 0 ºC.
	diTemperature	DINT	Temperatura lida no sensor interno da UCP., medida em graus Celsius, tendo sua resolução dada em 1°C.
Serial. COM[1-2].*	byProtocol	ENUM (BYTE)	Protocolo selecionado na COM 1: NO_PROTOCOL (0): Sem protocolo MODBUS_RTU_MASTER (1): MODBUS RTU Mestre MODBUS_RTU_SLAVE (2): MODBUS RTU Escravo OTHER_PROTOCOL (3): Outro protocolo
	dwRXBytes	DWORD	Contador de caracteres recebidos através da COM 1. (0 a 4294967295)
	dwTXBytes	DWORD	Contador de caracteres transmitidos através da COM 1. (0 a 4294967295)
	wRXPendingBytes	WORD	Número de caracteres pendentes no buffer de leitura na COM 1. (0 a 65535)
	wTXPendingBytes	WORD	Número de caracteres pendentes no buffer de transmissão na COM 1. (0 a 65535)
	wBreakErrorCounter	WORD	Esses contadores serão reiniciados nas seguintes condições: Energização Configuração da porta serial COM 1 Remoção das filas RX e TX Obs.: Quando a UCP estiver configurada com paridade Sem Paridade, o contador de erros de paridade não é incrementado caso
	wParityErrorCounter	WORD	
	wFrameErrorCounter	WORD	
	wRXOverrunCounter	WORD	

			receba uma paridade diferente. Nesse caso, será indicado erro de frame. O valor máximo de cada contador é 65535.
Ethernet.NET[1-6].*	bLinkDown	BOOL	Indica o estado do link na interface.
	byOperatingMode	ENUM (BYTE)	Modo de operação da interface: STANDALONE (0): Modo normal NIC_TEAMING (1): Modo NIC Teaming com outra interface SWITCH (2): Modo switch.
	byOperatingState	ENUM (BYTE)	Estado da interface: ERROR_UNCONFIGURED (0): Interface não configurada. DISABLED (1): Interface desabilitada. CONFIGURED_AND_RUNNING (2): Interface configurada e em funcionamento.
	wProtocol	WORD	Protocolo selecionado na NET 1: 00: Sem protocolo
	wProtocol. bMODBUS_RTU_ETH_Client	BOOL	Cliente MODBUS RTU via TCP
	wProtocol. bMODBUS_ETH_Client	BOOL	Cliente MODBUS TCP
	wProtocol. bMODBUS_RTU_ETH_Server	BOOL	Servidor MODBUS RTU via TCP
	wProtocol. bMODBUS_ETH_Server	BOOL	Servidor MODBUS TCP
	wProtocol.bDNP3_Client	BOOL	Cliente DNP3
	wProtocol.bDNP3_Server	BOOL	Servidor DNP3
	wProtocol.bIEC61850_Server	BOOL	Servidor 61850
	szIP	STRING(15)	Endereço IP da porta
	szMask	STRING(15)	Máscara de Subrede da porta
	szGateway	STRING(15)	Endereço Gateway da porta
	szMAC	STRING(17)	Endereço MAC da porta
	abyIP	BYTE ARRAY(4)	Endereço IP da porta
	abyMask	BYTE ARRAY(4)	Máscara de Subrede da porta
	abyGateway	BYTE ARRAY(4)	Endereço Gateway da porta
	abyMAC	BYTE ARRAY(6)	Endereço MAC da porta
	NICTeaming.* bStandbyState	BOOL	Interface está em estado de espera
	NICTeaming.* bActiveState	BOOL	Interface está em estado ativo
	NICTeaming.* bControllerFailure	BOOL	Falha no NIC Teaming
	NICTeaming.* bLinkDown	BOOL	Interface sem link
	NICTeaming.* bInterMsgTimeout	BOOL	Time-out na espera por pacotes do par NIC Teaming.
	NICTeaming.* bGeneralRxMsgTimeout	BOOL	Time-out na espera por pacotes da red.
	dwTransmittedBytes	DWORD	Contador de bytes enviados através da interface. (0 a 4294967295)
	dwTransmittedPackets	DWORD	Contador de pacotes enviados através da interface. (0 a 4294967295)
	dwTransmittedDropErrors	DWORD	Contador de perdas de conexão na transmissão através da interface. (0 a 4294967295)
	dwTransmittedCollisionErrors	DWORD	Contador de erros de colisão na transmissão através da interface. (0 a 4294967295)
	dwReceivedBytes	DWORD	Contador de bytes recebidos através da porta. (0 a 4294967295)
	dwReceivedPackets	DWORD	Contador de pacotes recebidos através da porta. (0 a 4294967295)
	dwReceivedDropErrors	DWORD	Contador de perdas de conexão na recepção através da interface. (0 a 4294967295)
	dwReceivedFrameErrors	DWORD	Contador de erros de frame na recepção através da interface. (0 a 4294967295)
UserFiles.*	byMounted	BYTE	Indica se a memória utilizada para gravar arquivos de usuário está apta para receber os dados.
	dwFreeSpacekB	DWORD	Espaço livre da memória de arquivos de usuário em Kbytes

	dwTotalSizekB	DWORD	Capacidade de armazenamento da memória de arquivos de usuário em Kbytes
UserLogs.*	byMounted	BYTE	Status da memória em que são inseridos os logs de usuário
	wFreeSpacekB	WORD	Espaço livre na memória de logs de usuário em Kbytes
	wTotalSizekB	WORD	Capacidade de armazenamento da memória de logs de usuário em Kbytes.
MemoryCard.*	byMounted	ENUM (BYTE)	Status do Cartão de Memória: NOT_MOUNTED (0): Cartão de memória não montado MOUNTED (1): Cartão de memória inserido e montado
	bMemcardtoCPUEnabled	BOOL	Nível de Proteção do Cartão de Memória: Leitura de dados do cartão de memória pela UCP autorizada.
	bCPUtoMemcardEnabled	BOOL	Escrita de dados no cartão de memoria pela UCP autorizada
	dwFreeSpacekB	DWORD	Espaço livre no Cartão de Memória em Kbytes
	dwTotalSizekB	DWORD	Capacidade de armazenamento do Cartão de Memória em Kbytes

WHSB.	byHotSwapAndStartupStatus	ENUM (BYTE)	Informa a situação anormal no barramento que ocasionou a parada da aplicação, para cada Modo de troca a quente: INITIALIZING (0): Esse estado é apresentado enquanto os demais estados não estiverem prontos. RESET_WATCHDOG (1): Aplicação em Modo Stop devido ao reset por cão-de-guarda de hardware ou por uma reinicialização do Runtime, quando desabilitada a configuração “Iniciar Aplicação de Usuário após Reset por Cão-de-Guarda”. ABSENT_MODULES_HOT_SWAP_DISABLED (2): Aplicação em Modo Stop devido à ativação do diagnóstico Módulos Ausentes, quando configurada troca a quente desabilitada. CFG_MISMATCH_HOT_SWAP_DISABLED (3): Aplicação em Modo Stop devido à ativação do diagnóstico Configuração Incompatível, quando configurada troca a quente desabilitada. ABSENT_MODULES_HOT_SWAP_ STARTUP_CONSISTENCY (4): Aplicação em Modo Stop devido à ativação do diagnóstico Módulos Ausentes, quando configurada troca a quente com consistência na partida somente para os módulos declarados. CFG_MISMATCH_HOT_SWAP_ STARTUP_CONSISTENCY (5): Aplicação em Modo Stop devido à ativação do diagnóstico Configuração Incompatível, quando configurada troca a quente com consistência na partida somente para os módulos declarados. APPL_STOP_ALLOWED_TO_RUN (6): Aplicação em Modo Stop e todas as consistências realizadas com sucesso. Aplicação pode ser colocada no Modo Run. APPL_STOP_MODULES_NOT_READY (7): Aplicação em Modo Stop e todas as consistências realizadas com sucesso, mas os módulos de E/S não estão aptos para a partida do sistema. Não é possível colocar a Aplicação em Modo Run. APPL_STOP_MODULES_GETTING_READY_TO_RUN (8): Aplicação em Modo Stop e todas as consistências realizadas com sucesso. Os módulos de E/S estão sendo preparados para a partida do sistema. Não é possível colocar a Aplicação em Modo Run. NORMAL_OPERATING_STATE (9): Aplicação em Modo Run. MODULE_CONSISTENCY_OK (10): Para uso interno. Todas as consistências foram realizadas com sucesso porém algum módulo não está em estado operacional. APPL_STOP_DUE_TO_EXCEPTION (11): Aplicação em modo Stop pois uma exceção ocorreu na UCP. DUPLICATED_SLOT_HOT_SWAP_DISABLED (12): Aplicação em Modo Stop devido à ativação do diagnóstico Slots Duplicados, quando configurada troca a quente desabilitada. DUPLICATED_SLOT_HOT_SWAP_STARTUP_CONSISTENCY (13): Aplicação em Modo Stop devido à ativação do diagnóstico Slots Duplicados, quando configurada troca a quente com consistência na partida somente para os módulos declarados. DUPLICATED_SLOT_HOT_SWAP_ENABLED (14): Aplicação em Modo Stop devido à ativação do diagnóstico Slots Duplicados, quando configurada troca a quente habilitada sem consistência na partida. NON_DECLARED_MODULE_HOT_SWAP_STARTUP_CONSISTENCY (15): Aplicação em Modo Stop devido à ativação do diagnóstico Módulo Não Declarado, quando configurada troca a

			quente habilitada com consistência na partida. NON_DECLARED_MODULE_HOT_SWAP_DISABLED (16): Aplicação em Modo Stop devido à ativação do diagnóstico Módulo Não Declarado, quando configurada troca a quente desabilitada.

	adwRackIOErrorStatus	DWORD ARRAY(32)	Identificação de erros em módulos de E/S, individualmente: O Array[0..31] representa 32 bastidores, sendo cada posição composta por 32 bits. Cada bit destas DWORDs representa uma posição do barramento, sendo o Bit-0 equivalente a posição 0. Então, é realizada uma operação lógica OU entre quatro diagnósticos resumidos e o estado operacional do módulo de uma determinada posição, e se o resultado for positivo, o bit análogo será verdadeiro. Abaixo, os diagnósticos considerados: Configuração incompatível (bConfigMismatch) Módulos ausentes (bAbsentModules) Módulos trocados (bSwappedModules) Módulos com erro fatal (bModuleFatalError) Módulo está em estado não operacional
	adwModulePresenceStatus	DWORD ARRAY(32)	Status de presença de módulos declarados de E/S em barramentos, individualmente: O Array[0..31] representa 32 bastidores, sendo cada posição composta por 32 bits. Cada bit destas DWORD representa uma posição do barramento, sendo o Bit-0 equivalente a posição 0. Então, se o módulo estiver presente, o bit será verdadeiro. Obs.: Esse diagnóstico é válido para todos os módulos, exceto fontes de alimentação não redundantes, módulos não declarados e as UCPs, ou seja, não apresentam a presença no barramento em suas respectivas posições (bit permanece em falso).
	byWHSBBusErrors	BYTE	Contador de falhas no barramento WHSB. Esse contador é reinicializado na energização. (0 a 255)
Application.*	byCPUState	ENUM (BYTE)	Informa o estado de operação da UCP: RUN (1): A aplicação está em Modo Run. STOP (3): A aplicação está em Modo Stop.
	bForcedIOs	BOOL	Existem um ou mais pontos de E/S forçados.
Events	wUsage	WORD	Amostragem do uso da fila de eventos
	bEventQueueOverFlow	BOOL	Bit que indica que ocorreu OverFlow na fila de Eventos, veja o capítulo ClearRtuDiagnostic.
	bEventQueueCleared	BOOL	Bit que indica que a fila de eventos foi limpa, veja o capítulo ClearEventQueue.
	byReserved_00	BYTE	Reservado
	wReserved_01	WORD	
	dwReserved_02	DWORD	
TimeManager*.	eActiveTimeSource	ENUM (BYTE)	Indica qual é a fonte de sincronismo ativa: NONE (0): Nenhuma fonte de sincronismo está ativa IRIGB (1): Está ativa a sincronização de tempo via IRIGB SNTP (2): Está ativa a sincronização de tempo via SNTP DNP (3): Está ativa a sincronização de tempo via DNP3 Obs: este diagnóstico pode levar alguns segundos para ser atualizado após a ocorrência de troca da fonte de sincronismo ativa do sistema (o que também ocorre na partida do sistema).
TimeManager. SNTP. *	bServiceEnabled	BOOL	Serviço SNTP habilitado.
	byActiveTimeServer	ENUM (BYTE)	Indica qual servidor está ativo: NO_TIME_SERVER (0): Nenhum servidor ativo. PRIMARY_TIME_SERVER (1): Servidor primário ativo. SECONDARY_TIME_SERVER (2): Servidor secundário ativo.

	wPrimaryServerDownCount	WORD	Contador de vezes que o servidor primário esteve indisponível. (0 a 65535)
	wSecondaryServerDownCount	WORD	Contador de vezes que o servidor secundário esteve indisponível. (0 a 65535)
	dwRTCTimeUpdatedCount	DWORD	Contador de vezes que o RTC foi atualizado pelo serviço SNTP. (0 a 4294967295)
	byLastUpdateSuccessful	ENUM (BYTE)	Indica status da última atualização: NOT_UPDATED (0): Não foi atualizado. UPDATE_FAILED (1): Última atualização falhou. UPDATE_SUCCESSFUL (2): Última atualização teve sucesso.
	byLastUpdateTimeServer	ENUM (BYTE)	Indica qual servidor foi utilizado na última atualização: NO_TIME_SERVER (0): Nenhuma atualização. PRIMARY_TIME_SERVER (1): Servidor primário. SECONDARY_TIME_SERVER (2): Servidor secundário.
	sLastUpdateTime	EXTENDED_DATE_AND_TIME	Data e horário da última sincronização de tempo via SNTP.
	sLastUpdateTime.byDayOfMonth	BYTE	
	sLastUpdateTime.byMonth	BYTE	
	sLastUpdateTime.wYear	WORD	
	sLastUpdateTime.byHours	BYTE	
	sLastUpdateTime.byMinutes	BYTE	
	sLastUpdateTime.bySeconds	BYTE	
	sLastUpdateTime.byMilliseconds	WORD	
TimeManager. IRIGB. *	bServiceEnabled	BOOL	Serviço IRIGB habilitado.
	dwTimeUpdatedCount	DWORD	Contador de vezes que o horário foi atualizado pelo serviço IRIGB. (0 a 4294967295)
	byLastUpdateSuccessful	ENUM (BYTE)	Indica status da última atualização: NOT_UPDATED (0): Não foi atualizado. UPDATE_FAILED (1): Última atualização falhou. UPDATE_SUCCESSFUL (2): Última atualização teve sucesso.
	sLastUpdateTime	EXTENDED_DATE_AND_TIME	Data e horário da última sincronização de tempo via IRIGB.
	sLastUpdateTime.byDayOfMonth	BYTE	
	sLastUpdateTime.byMonth	BYTE	
	sLastUpdateTime.wYear	WORD	
	sLastUpdateTime.byHours	BYTE	
	sLastUpdateTime.byMinutes	BYTE	
	sLastUpdateTime.bySeconds	BYTE	
	sLastUpdateTime.byMilliseconds	WORD	
Rack.*	dwAbsentRacks	DWORD	Cada bit representa um numero de identificação de um bastidor, se algum bit for TRUE, isso significa que o bastidor, com o referido número de identificação, está ausente.
	dwDuplicatedRacks	DWORD	Cada bit representa um numero de identificação de um bastidor, se algum bit for TRUE, isso significa que mais de um bastidor está configurado com o mesmo número de identificação.
	dwNonDeclaredRacks	DWORD	Cada bit representa um numero de identificação de um bastidor, se algum bit for TRUE, significa que há algum bastidor configurado com um número de identificação que não está declarado no projeto.

Tabela 7-4. Descrição dos Diagnósticos Detalhados HX3040

Notas:

Visualização das Estruturas de Diagnóstico: As Estruturas de Diagnóstico adicionadas ao projeto podem ser visualizadas acessando o item “Library Manager” na treeview da janela do MasterTool Xtorm. Com isso, é possível visualizar todos os datatypes definidos na estrutura.

Contadores: Todos os contadores dos diagnósticos da UCP retornam a zero quando o seu valor limite é ultrapassado.

Variável de representação direta: O “n” representa o valor configurado na UCP, através do software MasterTool Xtorm, como endereço inicial de diagnósticos.

Código de exceção: O código de exceção gerado pelo RTS (Runtime System) pode ser consultado na Tabela 7-5:

Código	Descrição	Código	Descrição
0x0000	Não há exceção.	**0x0051**	Violação de acesso.
0x0010	Tempo do Cão-de-Guarda da tarefa IEC expirado(Cão-de-Guarda de Software).	**0x0052**	Instrução privilegiada.
0x0012	Erro na configuração dos IOs.	**0x0053**	Falha na página.
		0x0054	Overflow da pilha.
0x0013	Erro de Check-up após o download do Programa.	**0x0055**	Disposição inválida.
0x0014	Erro de rede de campo.	**0x0056**	Manobra inválida.
0x0015	Erro na atualização dos IOs.	**0x0057**	Página protegida.
0x0016	Tempo de ciclo(execução) excedido.	**0x0058**	Falha dupla.
0x0017	Atualização da aplicação Online é muito grande	**0x0059**	OpCode inválido.
0x0018	Referências externas não resolvidas.	**0x0100**	Desalinhamento do tipo de dado.
0x0019	Download foi rejeitado.	**0x0101**	Limite de arrays excedidos.
0x001A	Projeto não carregado, pois as variáveis retentivas não podem ser realocadas.	**0x0102**	Divisão por zero.
0x001B	Projeto não carregado e deletado.	**0x0103**	Overflow.
0x001C	Fora da pilha da memória.	**0x0104**	Não continuável.
0x001D	Memória retentiva corrompida ou não pode ser mapeada.	**0x0105**	Cão-de-guarda na carga do processador de todas as tarefas IEC detectadas.
0x001E	Projeto pode ser carregado, mas causa uma queda posteriormente.	**0x0150**	FPU: Erro não especificado.
0x0021	Target da aplicação de inicialização não corresponde ao target atual.	**0x0151**	FPU: Operando não normal.
0x0022	Erro nas tarefas escalonadas.	**0x0152**	FPU: Divisão por zero.
0x0023	Check-up do arquivo baixado não corresponde.	**0x0153**	FPU: Resultado inexato.
0x0024	Identidade retentiva não corresponde com a atual identidade da aplicação de inicialização.	**0x0154**	FPU: Operação inválida.
0x0025	Falha na configuração da tarefa IEC.	**0x0155**	FPU: Overflow.
0x0026	Aplicação está executando com target errado.	**0x0156**	FPU: Verificação da pilha.
0x0050	Instrução ilegal.	**0x0157**	FPU: Underflow.

Tabela 7-5. Códigos de Exceção RTS

Notas:

Reset por Brownout: O diagnóstico de reset por brownout somente será verdadeiro quando a alimentação da fonte exceder o limite mínimo exigido nas características técnicas da mesma, mantendo-se com tensão baixa, ou seja, sem sofrer uma interrupção. A UCP irá identificar a queda da

alimentação e indicará o diagnóstico de falha na alimentação. Quando a tensão for restabelecida, a UCP será reinicializada automaticamente e indicará o diagnóstico de reset por brownout.

Diagnósticos via Blocos Funcionais

Os blocos funcionais proporcionam a visualização de alguns parâmetros que não podem ser acessados de outra maneira. As três funções sobre diagnósticos avançados estão localizadas na biblioteca LibPlcStandard e estão descritas abaixo.

GetTaskInfo

Essa função retorna informações sobre uma tarefa de uma determinada aplicação.

Figura 7-8. Função GetTaskInfo

Na Tabela 7-6 são descritos os parâmetros que devem ser repassados à função para que ela retorne as informações da aplicação.

Parâmetros de entrada	Tipo	Descrição
pszAppName	POINTER TO STRING	Nome da aplicação
pszTaskName	POINTER TO STRING	Nome da tarefa
pstTaskInfo	POINTER TO ST_TASK_INFO	Ponteiro para receber informações da tarefa

Tabela 7-6. Parâmetros de Entrada

Os dados que a função retorna, através do ponteiro informado nos parâmetros de entrada, são os descritos na Tabela 7-7.

Parâmetros retornados	Tamanho	Descrição
dwCurScanTime	DWORD	Tempo de ciclo (execução) da tarefa com 1µs de resolução.
dwMinScanTime	DWORD	Tempo mínimo de ciclo da tarefa com 1µs de resolução.
dwMaxScanTime	DWORD	Tempo máximo de ciclo da tarefa com 1µs de resolução.
dwAvgScanTime	DWORD	Tempo médio de ciclo da tarefa com 1µs de resolução.
dwLimitMaxScan	DWORD	Tempo máximo de ciclo da tarefa antes de ocorrer o cão-de-guarda.
dwIECCycleCount	DWORD	Contador de ciclos IEC decorridos

Tabela 7-7. Parâmetros Retornados

Possíveis TYPE_RESULT:

- OK_SUCCESS: execução com sucesso
- ERROR_FAILED: a tarefa desejada não existe

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
    sAppName : STRING;
    psAppName : POINTER TO STRING;
    sTaskName : STRING;
```

```
    psTaskName : POINTER TO STRING;
    pstTaskInfo : POINTER TO ST_TASK_INFO;
    TaskInfo : ST_TASK_INFO;
    Info : TYPE_RESULT;
END_VAR
//ENTRADAS:
sAppName := 'Application';       //Variável recebe o nome da aplicação.
psAppName := ADR( sAppName);     //Ponteiro com o nome da aplicação.
sTaskName := 'MainTask';         //Variável recebe o nome da tarefa.
psTaskName := ADR( sTaskName);   //Ponteiro com o nome da tarefa.
pstTaskInfo := ADR( TaskInfo);   //Ponteiro que irá receber as informações
da tarefa.
//FUNÇÃO:
//Chamada da função.
Info := GetTaskInfo ( psAppName, psTaskName, pstTaskInfo);
//Variável 'Info' recebe possíveis erros da função.
```

Visor Gráfico

O visor gráfico disponível na UCP da Série Hadron Xtorm é uma importante ferramenta para o controle de processo, pois através dele podem ser reconhecidas as possíveis condições de erro, presença de componentes ou de diagnósticos ativos. Além disso, é através do visor gráfico que todos os diagnósticos, inclusive dos módulos de E/S, são exibidos ao usuário. Para maiores detalhes sobre a utilização do botão de diagnósticos e da visualização dos mesmos, consultar a seção One Touch Diag.

Na Figura 7-9, é possível visualizar todos os caracteres disponíveis no visor gráfico da UCP Hadron Xtorm e, a seguir, os seus respectivos significados.

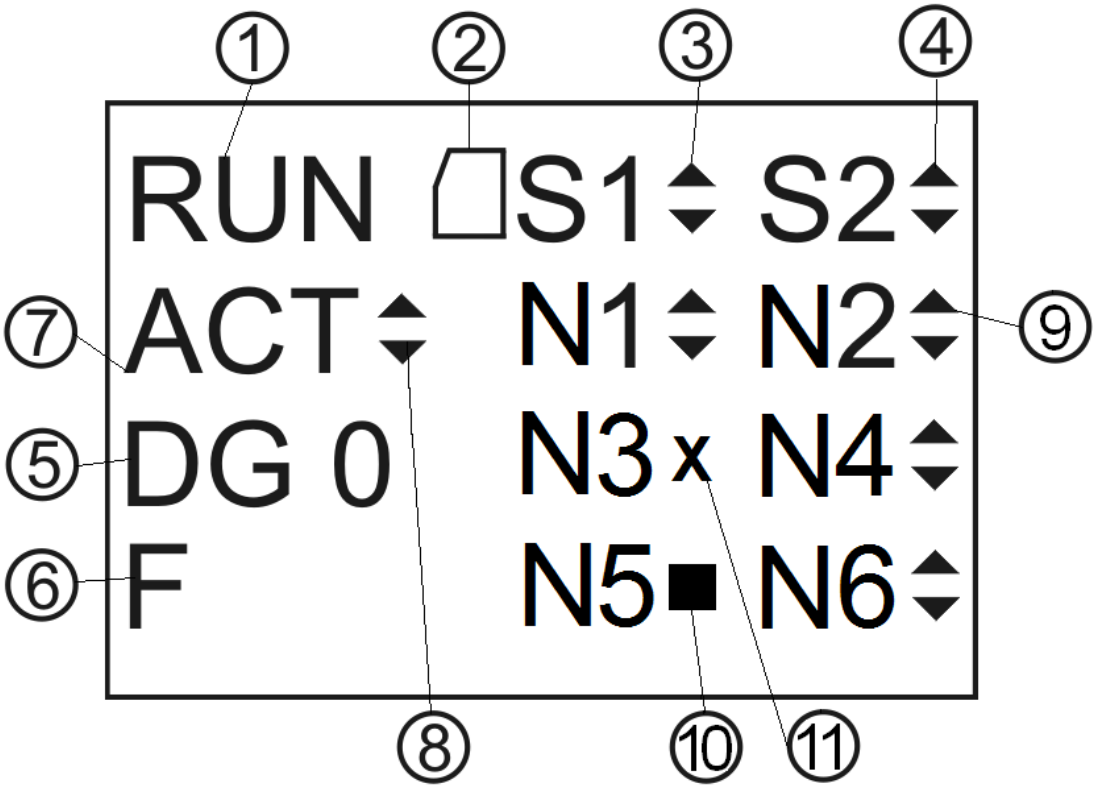


Figura 7-9. Tela de Status da UCP HX3040

Legenda:

1. Indicação do estado de operação da UCP. Caso a aplicação da UCP esteja em execução, o estado será Run. Caso a aplicação da UCP esteja parada, o estado será STOP e quando estiver parado

em marcas de depuração da aplicação o estado será BRKP. Para maiores detalhes, consultar a seção Estados de Operação da UCP

2. Indicação da presença do Cartão de Memória. Para maiores detalhes sobre a instalação do cartão, consultar o capítulo Instalação.
3. Indicação de Tráfego na COM 1 (S1). A seta para cima (▲) indica transmissão de dados e a seta para baixo (▼) indica recepção de dados. Para maiores informações sobre a interface COM 1, consultar seção Interfaces Seriais
4. Indicação de Tráfego na COM 2 (S2). A seta para cima (▲) indica transmissão de dados e a seta para baixo (▼) indica recepção de dados. Para maiores informações sobre a interface COM 2, consultar seção Interfaces Seriais
5. Indicação da quantidade de diagnósticos ativos na UCP. Caso o número mostrado seja diferente de 0 (zero), existem diagnósticos ativos na UCP. Para maiores detalhes sobre a visualização dos mesmos no visor gráfico da UCP, através do botão de diagnósticos, consultar seção One Touch Diag
6. Indicação de variáveis forçadas na UCP. Caso o caractere F esteja exibido no visor gráfico, alguma variável está sendo forçada pelo usuário, seja ela variável simbólica ou variável de representação direta mapeada em uma variável simbólica. Para maiores detalhes sobre forçamento de variáveis, consultar seção Modo Run
7. Identificação do estado da redundância na UCP (mensagem válida somente na HX3040 em modo redundante). Caso a UCP seja a UCP Ativa, a informação ACT será apresentada. Os demais estados possíveis são NCF (Não Configurado), STR (Inicializando), INA (Inativo) e SBY (Reserva).Para UCP simples e redundante as letra BRKP indicam (BreakPoint)
8. Indicação de que está sendo executada a sincronização de projeto. A seta para cima (▲) indica transmissão de dados do projeto e a seta para baixo (▼) indica recepção de dados de um projeto. Para maiores informações sobre a sincronização de projetos, consultar seção Sincronização de Projetos
9. Indicação de tráfego na porta Ethernet (N) (NET 1 ao NET 6). A seta para cima (▲) indica transmissão de dados e a seta para baixo (▼) indica recepção de dados. Para maiores informações sobre as interfaces Ethernet, consultar o capítulo Configuração das Interfaces Ethernet
10. Indicação de que a interface faz parte de um par NIC Teaming e que ela está em estado de reserva (parada). Lembrando que os pares de NIC Teaming serão sempre N1/N2, N3/N4 e N5/N6. Para maiores informações sobre os modos de configuração das interfaces Ethernet, consultar o capítulo de Configuração das Interfaces Ethernet
11. Indicação de ausência de link de rede (x) na porta Ethernet (N) (NET 1 ao NET 6). Para maiores informações sobre as interfaces Ethernet, consultar o capítulo Configuração das Interfaces Ethernet

Além dos caracteres descritos acima, a UCP Hadron Xtorm poderá apresentar algumas mensagens no visor gráfico, correspondentes a algum processo que está sendo executado no momento.

A Tabela 7-8 mostra as mensagens e suas respectivas descrições:

Mensagem	Descrição
FORMATANDO...	Indica que a UCP está formatando o cartão de memória.
ERRO NA FORMATACAO	Indica que ocorreu um erro durante a formatação do cartão de memória pela UCP.
FORMATO INCORR.	Indica que o formato do cartão de memória está incorreto.
SENHA INCORRETA	Indica que a senha digitada não confere com a senha configurada.
TRANSFERINDO...	Indica que o projeto está sendo transferido.
ERRO NA TRANSFER.	Indica que ocorreu um erro na transferência do projeto, ocasionado por algum problema no cartão de memória ou a remoção do mesmo durante a transferência.
TRANSFER. COMPLETA	Indica que a transferência foi completada com êxito.
TIMEOUT TRANSFER.	Indica que ocorreu um time-out (expirou tempo de comunicação) durante a transferência do projeto.

MODELO UCP INVALIDO	Indica que o modelo da UCP é diferente do configurado no projeto que está no cartão de memória.
VERSAO UCP INVALIDA	Indica que a versão da UCP é diferente do configurado no projeto que está no cartão de memória.
APLICACAO CORROMPIDA	Indica que a aplicação que está no cartão de memória está corrompida.
APLICACAO INEXISTENTE	Indica que não existe aplicação dentro do cartão de memória para ser transferida à UCP.
CRC INEXISTENTE	Indica que o CRC da aplicação não existe.
MCF INEXISTENTE	Indica que não existe o arquivo MCF dentro do cartão de memória.
SEM TAG	Não existe Tag configurada para a UCP no MasterTool Xtorm.
SEM DESC	Não existe Descrição configurada para a UCP no MasterTool Xtorm.
ERRO MSG.	Índica que há erro(s) na(s) mensagem(ns) de diagnóstico(s) do(s) módulo(s) requisitado(s).
SEM ASSINATURA	Indica que o produto apresentou um problema inesperado. Entre em contato com o setor de Suporte da Altus.
ERRO NA APL. REINICIANDO	Indica que ocorreu um erro na aplicação e o Runtime está reiniciando a aplicação.
APL. NAO CARREGADA	Indica que o Runtime não irá carregar a aplicação.
CARREGANDO APL.	Indica que o Runtime irá carregar a aplicação.
POSICAO INCORRETA	Indica que a UCP está em uma posição incorreta no bastidor.
ERRO FATAL	Indica que há problemas graves na inicialização da UCP como por exemplo, as partições da UCP não foram devidamente montadas. Entre em contado com o setor de suporte da Altus
HW-SW INCOMPATIVEL	Indica que o software e hardware da UCP não são compatíveis pois o produto apresentou algum problema inesperado. Entre em contato com o setor de suporte da Altus.
ATUALIZANDO FIRMWARE	Indica que o firmware está sendo atualizado na UCP.
RECEBENDO FIRMWARE	Indica que o arquivo de atualização está sendo transferido para a UCP.
ATUALIZADO:	Exibe a versão de firmware atualizada na UCP.
ERRO NA ATUALIZACAO	Indica que houve algum erro durante a atualização de firmware da UCP, ocasionado por falha na comunicação ou problemas de configuração.
REINICIANDO SISTEMA...	Indica que a UCP está sendo reiniciada para que as atualizações tenham efeito.

Tabela 7-8. Outras Mensagens do Visor Gráfico do HX3040

Log de Sistema

O Log de Sistema é um recurso disponível no programador MasterTool Xtorm. É uma importante ferramenta para o controle de processo, pois através dele é possível localizar eventos na UCP que podem indicar condições de erro, presença de componentes ou de diagnósticos ativos. Tais eventos podem ser visualizados em ordem cronológica com uma resolução de milissegundos, com uma capacidade de armazenamento de até mil entradas de Log. Para acessar esses Logs, basta ir na Árvore de Dispositivos e dar um duplo-click em Device, e em seguida ir na aba Log.

Para que os Logs possam ser visualizados, basta estar conectado a uma UCP (Caminho Ativo selecionado) e clicar em .

É possível filtrar os Logs em 4 tipos diferentes: Advertência(s), Erro(s), Exceção (s),Informação (s).

Outra maneira de filtrar as mensagens exibidas para o usuário é selecionar qual o componente queremos visualizar.

Deve-se atentar ao fato de que a Marca de Tempo dos Logs será sempre de acordo com o UTC.

Nota

Log de Sistema nas UCPs: Os logs de sistema da UCP da série Hadron Xtorm não são recarregados no caso de uma reinicialização da UCP ou por uma reinicialização do Runtime, isto é, este log não é retentivo, não será possível visualizar os logs mais antigos quando uma dessas condições ocorrer.

Diagnósticos dos Módulos de E/S

As funções de acesso do tag, Descrição e Diagnósticos dos módulos e pontos de E/S estão intimamente relacionadas entre si. Existem três componentes principais que serão amplamente referências nesta explicação, são eles:

1. Visor gráfico da UCP
2. Visor do módulo a ser acessado
3. Botão de diagnóstico do módulo a ser acessado

Acesso ao modo de diagnóstico

Independentemente do que está sendo mostrado no visor da UCP, após um pressionamento curto no botão de diagnóstico de um determinado módulo, será mostrado no visor da UCP o tag e os diagnósticos ativos do respectivo módulo. Estes dados serão mostrados no visor da UCP na ordem indicada na Figura 7-10.

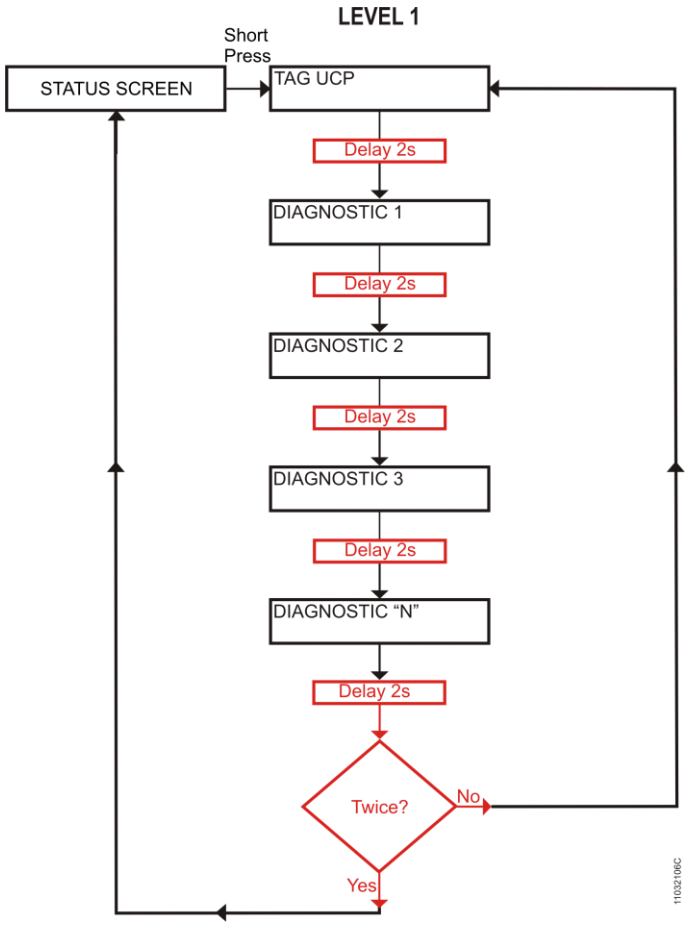


Figura 7-10. Visualização dos diagnósticos da UCP

Conforme mostrado na Figura 7-10, tanto o tag quanto a lista de todos os diagnósticos ativos relacionados ao módulo serão mostrados duas vezes no visor da UCP, após, o respectivo módulo sairá do modo diagnóstico e o visor da UCP voltará a indicar informações referentes à UCP.

É possível identificar que um determinado módulo está em modo diagnóstico quando os segmentos, indicados na Figura 7-11, do visor do módulo estiverem piscando. A quantidade de segmentos depende da quantidade de pontos de cada um dos módulos de E/S.

Figura 7-11. Visor do Módulo

Identificando pontos de E/S com diagnósticos individuais

Após entrar no modo diagnóstico, os módulos de E/S indicarão quais pontos que possuem diagnósticos individuais, piscando o segmento respectivo ao determinado ponto de E/S. Por exemplo, se o segmento indicado na Figura 7-12 estiver piscando, indicará que o segundo ponto de E/S do módulo possui algum diagnóstico individual ativo. Tal funcionalidade se torna útil para a detecção rápida de diagnósticos individuais.

Figura 7-12. Exemplo de Diagnóstico

Acessando pontos de E/S

Após entrar no modo diagnóstico, o próximo pressionamento curto do botão de diagnósticos do módulo em questão selecionará o primeiro ponto de E/S, neste momento o visor apagará a indicação de diagnósticos individuais ativos (explicado anteriormente) e indicará qual é o ponto de E/S que está selecionado. Para a seleção do próximo ponto de E/S basta um novo pressionamento curto no botão de diagnósticos do respectivo módulo. Quando o último ponto de E/S estiver selecionado um novo pressionamento curto no botão de diagnóstico irá finalizar o modo de diagnóstico.

Ao acessar um ponto de E/S, o visor da UCP indicará o tag do respectivo ponto de E/S e todos os diagnósticos ativos do mesmo da mesma forma como indicada no fluxograma da Figura 7-10

Para módulo que não possuem pontos de E/S, um novo pressionamento curto no botão de diagnósticos, após a entrada no modo diagnóstico, finalizara a operação.

Acessando a descrição do módulo e dos pontos de E/S

Além do tag, os módulos e pontos de E/S podem possuir uma descrição. Indica-se utilizar a descrição quando existe a necessidade de adicionar uma informação extra, além do tag para um determinado módulo ou ponto de E/S como, por exemplo: “Módulo de leitura de temperaturas” para a descrição de um módulo ou “Acionamento bomba principal” para a descrição de um determinado ponto de saída. O limite de caracteres para o tag, descrição da tag ou nome do módulo é de 255 caracteres.

Para alterar o nome e a descrição de cada módulo inserido na aplicação, clique com o botão direito sobre o módulo, no item “Propriedades”, na guia “Comum”, altere o nome ou a descrição sendo ambos limitados a 255 caracteres.

Para alterar a tag, acesse a aba “Bus I/O Mapping” de cada módulo, e dê um duplo clique na coluna da tabela destinada à inserção da tag de cada ponto. Para alterar a descrição da tag, faça o mesmo na coluna destinada para descrição da tag, e insira a informação desejada.

Para acessar a descrição, basta um pressionamento longo no botão de diagnóstico quando o módulo estiver indicando o tag ou os diagnósticos ativos do módulo. Nesse caso aparecerá no visor da UCP a descrição do módulo. Da mesma forma, quando estiver selecionado um ponto de E/S e for feito um pressionamento longo no botão de diagnósticos, aparecerá no visor da UCP a descrição do determinado ponto de E/S.

ATENÇÃO:
É recomendável que a tag, e o nome do módulo possuam apenas os caracteres alfanuméricos (maiúsculos ou minúsculos e o texto não iniciar com números). Para a descrição o recomendável é os caracteres alfanuméricos (maiúsculos ou minúsculos), o espaço em branco e o caractér ponto “.”. A utilização de qualquer outro caractér diferente dos acima citados não é recomendada.

ATENÇÃO:
Quando utilizar a função ETD – Electronic Tag on Display, Diagnostic Explorer ou Web Server para visualizar a tag dos pontos de E/S, o nome da tag será truncado nos primeiros 24 caracteres após a string “Application.” do nome da tag. Ex. Para o tag "Application.UserPrg.MyTest.ON", somente a string “UserPrg.MyTest.ON" será visualizada no visor gráfico.

ATENÇÃO:
Quando utilizar a função ETD – Electronic Tag on Display, Diagnostic Explorer ou Web Server para visualizar a descrição do tag dos pontos de E/S, a descrição da tag será truncada nos primeiros 48 caracteres da descrição dada para o tag.

ATENÇÃO:
Quando utilizar a função ETD – Electronic Tag on Display, Diagnostic Explorer ou Web Server para visualizar o nome do módulo, o nome do módulo será truncado nos primeiros 24 caracteres da descrição dada para o nome do módulo.

Pressionamento curto e pressionamento longo

A Tabela 7-9 indica os tempos

Tipo Pressionamento	Tempo Mínimo	Tempo Máximo	Condição para Indicação
Sem pressionamento	-	59,99 ms	-
Pressionamento	60 ms	0,99 s	Pressionamento e

curto			liberação da tecla dentro do período definido
Pressionamento longo	1 s	20 s	Pressionamento por mais de 1 segundos
Tecla Presa	20,01 s	∞	Pressionado por mais de 20 segundos

Tabela 7-9. Tempos de Pressionamento Botão de diagnóstico

Não Carregar a Aplicação na Inicialização

Caso seja necessário, o usuário pode optar por não carregar uma aplicação já existente na UCP durante sua inicialização. Para isto, basta energizar a UCP com o botão de diagnóstico pressionado e mantê-lo assim por, pelo menos, dois segundos. Durante a inicialização da UCP será exibida no visor uma mensagem avisando que a aplicação não será carregada e inicializará em Modo Stop. Caso seja feito um Login, o software MasterTool Xtorm indicará que não existe nenhuma aplicação na UCP. Para voltar a carregar a aplicação, a UCP deve ser reinicializada ou um novo download da aplicação deve ser feito.

Falha na Alimentação

A fonte de alimentação da Série Hadron Xtorm possui um sistema de detecção de falha, de acordo com os níveis definidos em suas características técnicas (consultar Características Técnicas do Módulo de Alimentação).

1 – Caso a fonte seja energizada com tensão inferior ao limite mínimo exigido, será gerado um diagnóstico de falha de alimentação, o qual será reconhecido pela UCP e a mensagem "FALHA NA ALIMENTAÇÃO" será exibida em seu visor. Quando a alimentação estiver dentro dos limites estabelecidos, a UCP irá reconhecer e, automaticamente, será reiniciada com a aplicação do usuário. O diagnóstico ainda estará ativo para mostrar ao usuário que na última inicialização ocorreu falha na alimentação.

2 – Caso a fonte tenha uma queda de tensão para um valor inferior ao limite mínimo exigido e retornar para um valor acima em até 10 ms, a falha de alimentação não será reconhecida pela UCP e não será gerado o diagnóstico, pois o sistema se mantém intacto durante esse tempo.

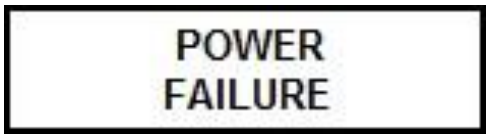


Figura 7-13. Mensagem de Falha na Alimentação

Problemas mais Comuns

Se, ao energizar a UCP, a mesma não entra em funcionamento, os seguintes itens devem ser verificados:

- A temperatura ambiente está dentro da faixa suportada pelos equipamentos?
- A fonte de alimentação do bastidor está sendo alimentada com a tensão correta?
- A fonte de alimentação é o módulo mais a esquerda no bastidor (bastidor sendo visto de frente), e a sua direita é seguido pela UCP da Série Hadron Xtorm? Caso a arquitetura utilizada seja com fontes redundantes, o módulo mais a esquerda no bastidor será a fonte, a sua direita é seguido por outra fonte, e na próxima posição a direita deverá ser UCP da Série Hadron Xtorm.
- Os equipamentos da rede, como hubs, switches ou roteadores, estão alimentados, interligados, configurados e funcionando corretamente?

- O cabo de rede Ethernet está devidamente conectado à porta NET 1 a NET 6 da UCP Hadron Xtorm e ao equipamento de rede?
- A UCP da Série Hadron Xtorm está ligada, em modo de execução (Run) e sem diagnósticos relacionados à hardware?

Se a UCP Hadron Xtorm indica o estado execução (Run), mas não responde às comunicações solicitadas, sejam elas pelo MasterTool Xtorm ou através de protocolos, os seguintes itens devem ser verificados:

- A configuração dos parâmetros Ethernet da UCP estão corretos?
- O respectivo protocolo de comunicação está configurado corretamente na UCP?
- As variáveis que habilitam as relações MODBUS / DNP3 estão devidamente habilitadas?

Se nenhum problema for identificado, consulte o Suporte a Clientes Altus.

Solução de Problemas

A Tabela 7-10 mostra os sintomas de alguns problemas com suas possíveis causas e soluções. Se o problema persistir, entre em contato com o Suporte Técnico da Altus.

Sintoma	Possível Causa	Solução
Não liga	Falta de alimentação ou alimentado incorretamente.	Verificar se a UCP está conectada corretamente no bastidor. Desernegizar e retirar todos os módulos do barramento, menos a fonte de alimentação e a UCP. Energizar o barramento e verificar o funcionamento da fonte de alimentação, tanto a externa como a fonte conectada ao barramento. Verificar se a tensão de alimentação chega ao borne da fonte de alimentação Hadron Xtorm, e também se a polarização está correta.
Não comunica	Mau contato ou mal configurado.	Verificar todas as conexões dos cabos de comunicação. Verificar as configurações das interfaces serial e Ethernet no software MasterTool Xtorm.
Não reconhece o cartão de memória	Mal conectado ou desmontado	Verificar se o cartão de memória está conectado corretamente no compartimento, . Verificar se o cartão de memória foi colocado do lado correto, conforme a indicação no painel frontal da UCP. Verificar se o cartão de memória não foi desmontado através do menu específico na UCP, visualizando a indicação no visor gráfico da UCP.

Tabela 7-10. Tabela de Solução de Problemas

Manutenção Preventiva

- Deve-se verificar, a cada ano, se os cabos de interligação estão com as conexões firmes, sem depósitos de poeira, principalmente os dispositivos de proteção.
- Em ambientes sujeitos a contaminação excessiva, deve-se limpar periodicamente o equipamento, retirando resíduos, poeira, etc.
- Os diodos TVS utilizados para a proteção contra transientes causados por descargas atmosféricas devem ser inspecionados periodicamente, pois podem estar danificados ou destruídos caso a energia absorvida esteja acima do limite. Em muitos casos, a falha pode não ser evidente ou facilmente visível. Em aplicações críticas, é recomendável a substituição periódica dos diodos TVS, mesmo os que não apresentarem sinais visíveis de falha.
- Aperto e limpeza do barramento a cada 6 meses.

8. Projeto do Painel Elétrico

Projeto Mecânico

Dimensões

A dimensão de cada módulo da Série Xtorm pode ser encontrado no documento de Características Técnicas de cada módulo, cujos documentos estão listados na Tabela 1-1. Documentos Relacionados

Profundidade do Módulo Montado no Bastidor

A profundidade do conjunto módulo da Série Xtorm e bastidor da Série Xtorm pode ser encontrada ao somar 28 mm à profundidade do módulo. No exemplo da Figura 8-1 foi utilizado um módulo padrão com profundidade de 184,2 mm. Ao considerar o bastidor, a profundidade do conjunto passa para 212,2 mm podendo variar 0,5mm.

Dimensão em mm.

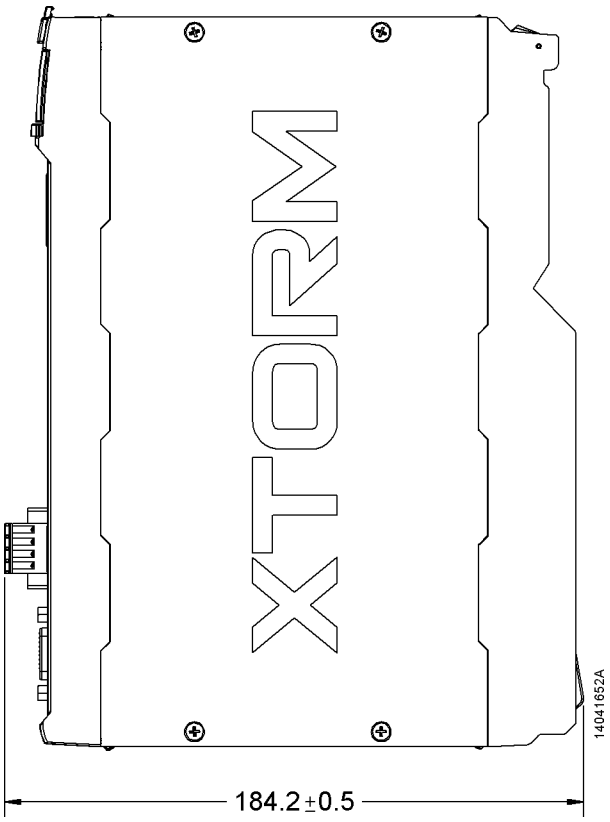


Figura 8-1. Profundidade do Módulo Montado no Bastidor

Espaçamento entre módulos e outros equipamentos do painel

A UTR necessita de espaços livrem em torno do mesmo. Isto é necessário para permitir o correto manuseio do equipamento. Além disto, tal espaçamento deve ser respeitado de forma a permitir a passagem de ar através da UCP, na forma de convecção, responsável por manter a temperatura do equipamento. A Figura 8-2 e a Tabela 8-1 indicam o espaçamento necessário para os módulos da Série Hadron Xtorm.

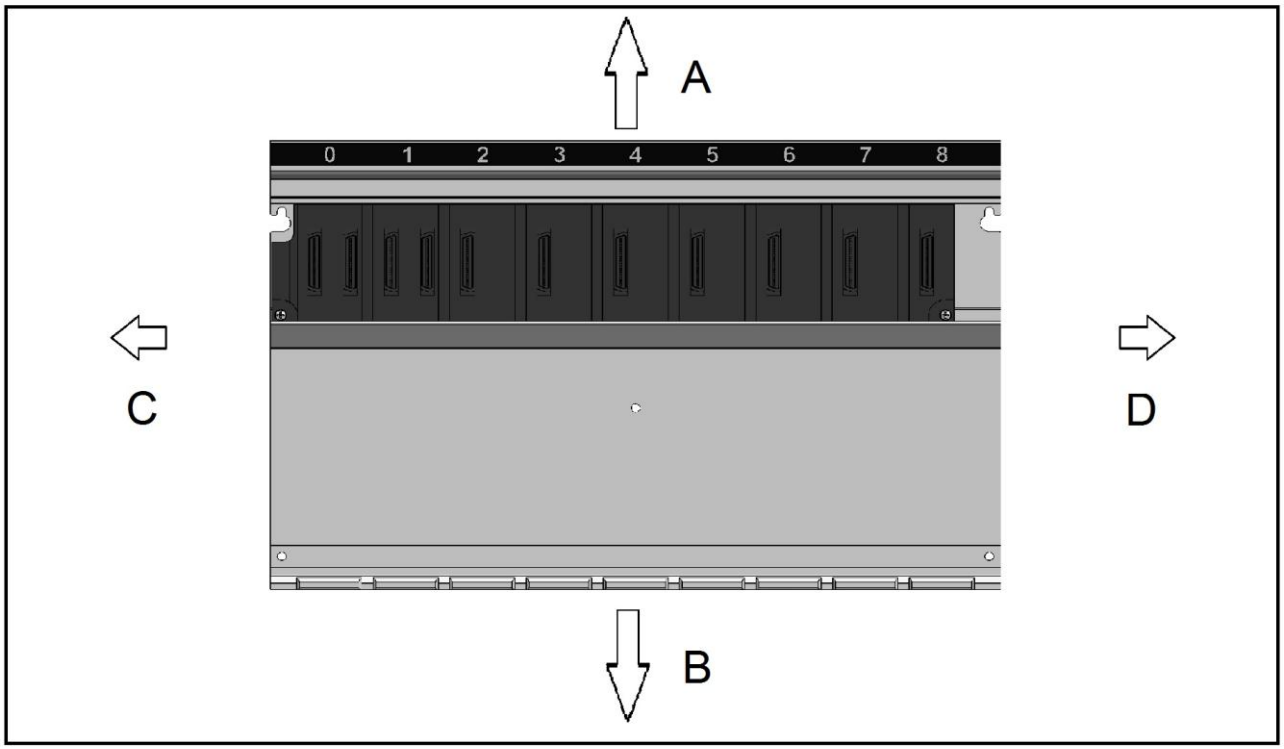


Figura 8-2. Espaço Livre em Torno da UTR

Dimensão A	Dimensão B	Dimensão C	Dimensão D
10 cm	10 cm	4 cm	4 cm

Tabela 8-1. Dimensões do Espaço Livre em Torno da UTR

Dimensionamento da Calha

Para o dimensionamento da calha, além da área ocupada pelos fios, observar o aquecimento interno na calha, provocado pelo calor dissipado pelos fios, que pode levar a uma redução na área de ocupação da calha.

Utilizar a seguinte regra: área da calha >= somatório da área dos fios / 0,4

Área dos fios = (3,14 * raio ^ 2)

Considera-se como área dos fios a área total, incluindo a isolação.

Montagem Horizontal/ Vertical

A Série Xtorm permite a utilização da UTR na orientação horizontal. Não é permitida a montagem vertical do bastidor.

Projeto Térmico

Os equipamentos Altus são projetados para trabalhar a uma temperatura ambiente de até 60°C (exceto quando especificado). Portanto, esta deve ser a temperatura interna máxima do armário. Os seguintes cuidados devem ser observados no projeto do painel:

Dimensionar armários com volume interno suficiente para uma boa circulação de ar

Prever ventilação forçada ou trocadores de ar com o meio externo, caso necessário, para que não haja elevação da temperatura além do limite. Em casos críticos, recomenda-se o uso de equipamentos de refrigeração, para manter o equipamento operando dentro dos níveis de temperatura de operação

Distribuir de forma homogênea fontes de calor dentro do armário

Considerar a dissipação nos cabos que conduzem correntes mais elevadas para evitar superaquecimento interno às calhas

ATENÇÃO Para obter a máxima dissipação de cada módulo da Série Xtorm, consulte o documento Característica Técnica do módulo em questão.

A seguir, é explicado um método para calcular a temperatura interna do painel em função da sua dissipação e potência.

Dissipação de calor em um painel elétrico

Cada painel elétrico dissipa, através de sua superfície, uma quantidade definida de calor para uma dada diferença de temperatura interna e externa. Para o cálculo da dissipação de calor em situações em que a diferença de temperatura entre o interior e o exterior do painel chegue até 50 °C, as seguintes grandezas devem ser consideradas:

Superfície de dissipação efetiva do painel; calculada segundo a norma DIN-VED 0660 capítulo 500, conforme indicado pelo tipo de instalação

A constante de dissipação para a chapa de aço pintada em W/m² °C

As condições de ventilação do painel (local de instalação)

Grau de ocupação interna do painel (impedância à circulação do ar no interior)

Dos valores citados anteriormente, apenas o valor da superfície do painel pode ser calculada exatamente.

Cálculo da superfície efetiva de dissipação A (m²) de um painel:

O cálculo da superfície “A” é feito conforme indicado pela norma DIN-VDE, segundo o tipo de instalação do painel:

Tipo de instalação conforme a norma DIN-VDE 0660/500	Fórmula para o cálculo de A (m²)
Painel livre de todos os lados	A = 1,8 * H * (L + P) + 1,4 * L * P
Painel com a superfície traseira obstruída	A = 1,4 * L * (H + P) + 1,8 * P * H
Painel com uma superfície lateral obstruída	A = 1,4 * L * (H + L) + 1,8 * L * H
Painel com uma lateral e a superfície traseira obstruídas	A = 1,4 * H * (L + P) + 1,4 * L * P
Painel com as duas laterais obstruídas	A = 1,8 * L * H + 1,4 * L * P + P * H
Painel com as duas laterais e a superfície traseira obstruídas	A = 1,4 * L * (H + P) + P * H
Painel com as duas laterais e a superfície traseira e superior obstruídas	A = 1,4 * L * H + 0,7 * L * P + P * H

Tabela 8-2.Cálculo da Superfície Efetiva de Dissipação

L = Largura (m), H = Altura (m), P = Profundidade (m)

Na aplicação em painéis construídos com chapa de aço pintada, para o ar parado ao seu redor, a constante de dissipação de calor pode ser considerada 5,5 W/m2² °C.

A potência dissipada por um painel pode então ser calculada por meio da equação Qs = k * A * (temperatura interna –temperatura externa), ou obtida a partir da Figura 8-3.

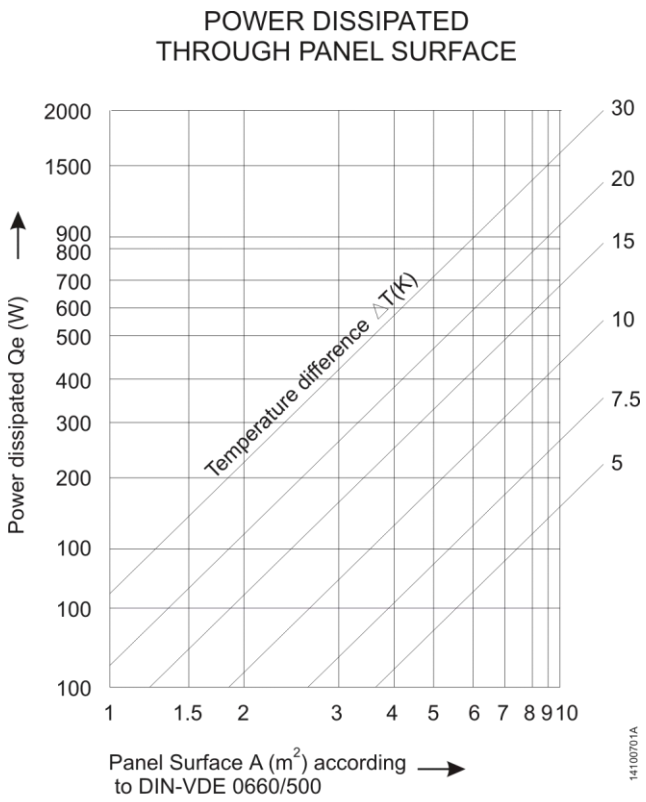


Figura 8-3. Potência Dissipada x Superfície x Dif. de Temperatura

Este valor poderá, no entanto, ser triplicado se for provocada circulação de ar no exterior do painel.

A circulação de ar em um painel é obstruída pela instalação dos equipamentos em seu interior, conduzindo à formação de focos de aquecimento localizados. Nesta situação, pode-se obter auxílio para a circulação desejada do ar por meio da instalação de ventiladores internos ao painel, aumentando o fluxo do ar em seu interior.

A circulação forçada através de ventiladores no interior do painel traz também uma melhora na convecção própria e uma tendência a se igualarem as temperaturas ao longo do painel. Sem a circulação forçada do ar tem-se um foco de calor no alto do painel, em virtude da convecção.

Exemplos:

Para um painel livre de todos os lados, com área efetiva de 3,96 m², potência instalada de 350 W e temperatura ambiente externa de 30 °C, calcule a temperatura média interna.

Qs = k * A * (Ti – Te)

350 = 5,5 * 3,96 * (Ti – 30)

Ti = 46 °C

Para o mesmo painel, calcule a temperatura interna para uma potência instalada de 1000 W.

Qs = k * A * (Ti – Te)

1000 = 5,5 * 3,96 * (Ti – 30)

Ti = 76 °C,

Neste caso, a temperatura excedeu o limite de operação dos equipamentos (60 °C), e deve ser providenciada uma outra maneira para retirada do calor excedente. O limite da potência instalada para a temperatura interna de 60 graus é:

Qs = k * A * (Ti – Te)

Qs = 5,5 * 3,96 * (60 – 30)

Qs = 653 W, sendo o limite 653 W, os 347 W restantes (1000 W – 653 W) devem ser retirados, por exemplo, através de um equipamento de ar-condicionado.

ATENÇÃO: Nos cálculos anteriores, observar que a temperatura interna, é sempre uma temperatura média, e que caso não haja circulação forçada de ar no interior do painel, a temperatura no topo do painel será maior que na base, e poderão existir focos quentes localizados. A devida margem de segurança deve ser dada em cada caso.

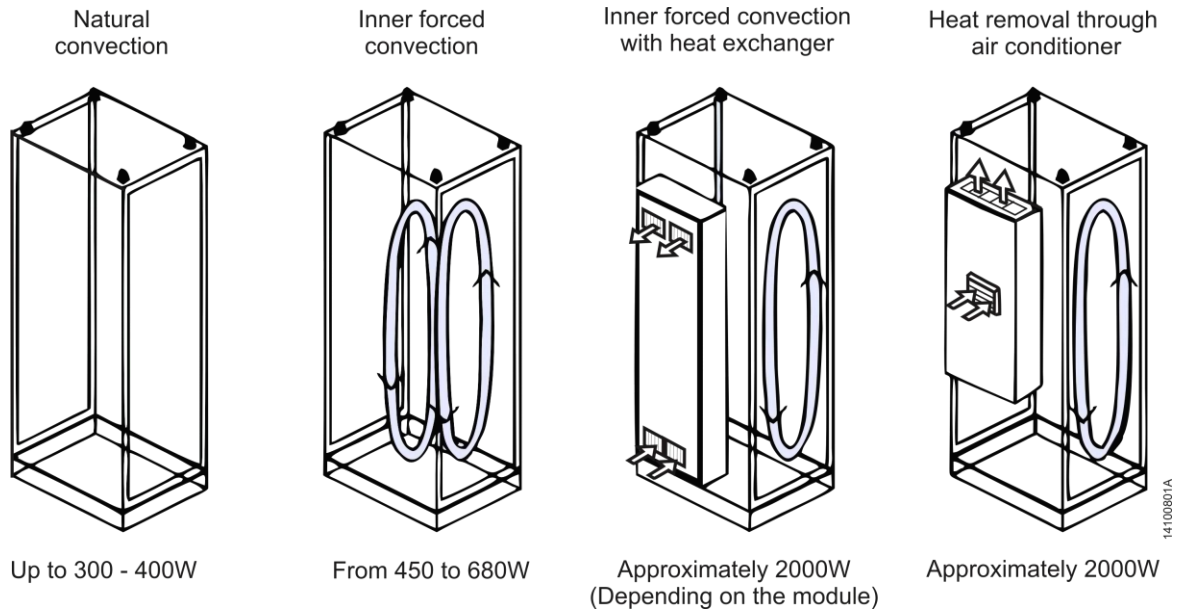


Figura 8-4. Exemplos de Movimentação do Calor – Instalação Fechada

Uma dissipação bem maior de calor, comparando-se com a obtida anteriormente, pode ser alcançada se for permitida a troca de ar com o exterior. A ventilação é normalmente realizada introduzindo-se venezianas de ventilação nas laterais, na porta ou na tampa traseira. Isto irá evidentemente reduzir o grau de proteção (IP) do painel.

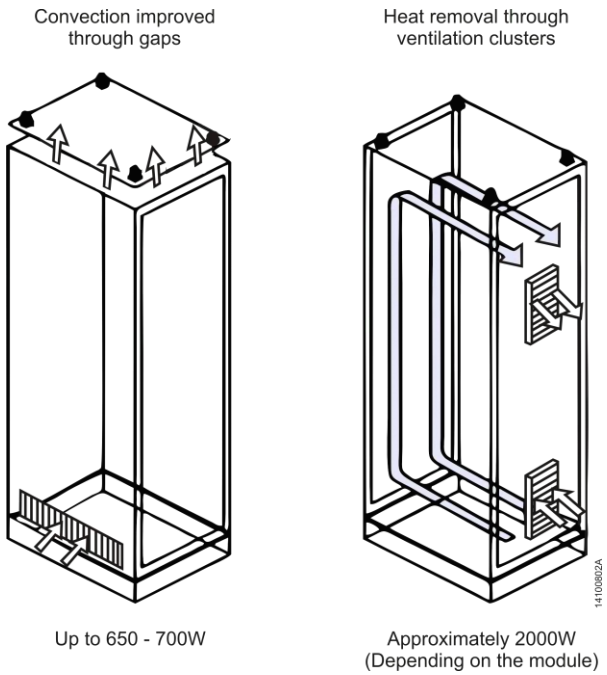


Figura 8-5. Exemplo de Movimentação do Calor – Instalação Aberta

Projeto Elétrico

Informações Gerais

As Unidades Terminais Remotas são fabricadas atendendo normas mundiais, que estabelecem os níveis aceitáveis de condições ambientais e de ruído normalmente encontrados em aplicações de controle de sistemas de geração, transmissão e distribuição de energia elétrica, como usinas hidrelétricas (UHEs), e subestações de energia.

É fundamental também que a instalação destes produtos siga regras de projeto convenientes, estabelecidas por normas de instalação. Problemas causados por interferências eletromagnéticas (EMI), tais como falhas de comunicação, falhas de execução de programa, ruído em variáveis analógicas, e até mesmo perda de programa, podem ser causadas por um projeto elétrico ou instalação deficientes.

O projeto elétrico das UTR´s Altus deve respeitar a norma IEEE 518/1977, "Guide for Installation of Electrical Equipment to Minimize Electrical Noise Inputs to Controllers External Sources". A seguir, os pontos mais importantes são abordados.

Alimentação do Armário

A alimentação do sistema de controle deve possuir chave geral. Recomenda-se o uso de bornes para alimentação geral do painel de montagem com fusíveis integrados, bem como a previsão de uma tomada fornecendo 127 ou 220 Vac, para uso do terminal de programação. É importante que esta tomada possua pino de aterramento, pois o terminal de programação deverá, obrigatoriamente, possuir conexão com o terra do sistema. Todas as tomadas do armário devem possuir indicação clara de suas tensões.

Distribuição dos Cabos no Armário

A correta distribuição dos cabos no armário e o correto aterramento das partes garantem a compatibilidade eletromagnética (EMC) da instalação.

É importante que as alimentações do painel elétrico sejam corretamente distribuídas, através de barras de distribuição ou bornes de ligação.

A partir destes pontos de distribuição geral, leva-se um cabo próprio a cada ponto específico a ser alimentado. Deve-se evitar ramificações locais nas alimentações dos módulos, diminuindo-se assim os percursos dos cabos conduzindo alta corrente.

Para o melhor desempenho do equipamento, é necessário separar os circuitos quanto ao seu tipo, para reduzir interferências eletromagnéticas, como segue:

1. Circuitos de alimentação AC e acionamentos de cargas AC e DC
2. Circuitos de entrada e saídas digitais de baixa corrente
   (menor ou igual a 1A)
3. Circuitos analógicos e de comunicação

Estes circuitos devem ser distribuídos preferencialmente em calhas separadas ou evitando-se que se disponham paralelamente uns aos outros. A distância mínima de 150 mm é recomendada entre todos os sinais de E/S e alimentações maiores que 500 V.

Iluminação do Armário

É fundamental que se coloque iluminação interna no armário, acionada por interruptor, para facilitar a sua operação.

Recomenda-se que a iluminação seja com lâmpadas incandescentes, pois lâmpadas fluorescentes podem gerar interferências indesejáveis. Se estas forem utilizadas, as seguintes precauções devem ser tomadas de modo a reduzir a interferência:

1. Colocar tela metálica aterrada entre a lâmpada e o armário, para reduzir a emissão de ruídos
2. Colocar blindagem nos cabos de alimentação da lâmpada
3. Proteger o interruptor em caixa metálica e colocar filtro na rede de alimentação preferencialmente junto à lâmpada

Aterramento

É necessária uma borneira geral ou uma barra de terra no armário, onde serão realizados todos os aterramentos de fontes e módulos. Esta barra deve estar ligada a um terra com baixa resistência.

Interferência Eletromagnética

A interferência eletromagnética (EMI) é responsável pela grande maioria dos problemas encontrados em equipamentos instalados.

Pode-se reduzir significativamente estes problemas se as seguintes precauções forem tomadas na fase de projeto do painel:

1. Distribuir e arranjar os cabos nas calhas, evitando misturar cabos de alimentação com cabos de sinais
2. Partes metálicas inativas devem ser aterradas no armário
3. Caso existam elementos que causem emissão de ruídos, recomenda-se a utilização de blindagens
4. Filtrar a entrada de alimentação do painel

Blindagem

Fortes fontes geradoras de interferência eletromagnética (transformadores, motores, cabos com alta corrente ou tensão) situadas dentro do armário, devem ser cobertas por chapas metálicas aterradas, quando situados a menos de 50 cm das partes eletrônicas da UTR. Cabos que ultrapassam as partes blindadas devem ser blindados ou filtrados.

Os cabos blindados dentro do armário devem ser aterrados conforme as instruções de cada equipamento.

Supressores de Ruído

É extremamente importante a conexão de supressores de ruído de porte adequado diretamente em todas as cargas indutivas (relés, contactoras, solenóides, etc.) acionadas ou não pela UTR. O acionamento de cargas indutivas gera fortes ruídos elétricos que podem ultrapassar os limites estabelecidos pelas normas. Os ruídos, se não atenuados em sua origem, podem atingir a UTR, afetando seu funcionamento.

Os circuitos de proteção devem ser montados próximos da carga, como regra, não devem estar afastados mais que 0,5 metros. No caso de cargas resistivas (lâmpadas incandescentes, LEDs de sinalização, resistores de aquecimento, etc.), não é necessário o uso destes dispositivos.

Circuito RC

O circuito de proteção RC (Resistor em série com um Capacitor) pode ser montado em paralelo com o contato ou em paralelo com a carga. A montagem em paralelo com os contatos é recomendada para cargas alimentadas em tensão contínua. A montagem em paralelo com a carga é recomendada para cargas alimentadas com tensões contínuas ou alternadas. Os circuitos RC são mais eficazes quando utilizados em tensões acima de 100 V.

Para selecionar os valores de R e C, recomenda-se que o resistor tenha de 0,5 a 1 Ohm para cada 1 V de tensão, e o capacitor tenha 0,5 a 1 μF para cada 1 A de corrente. Por exemplo, em uma carga de 220 V/1 A pode-se utilizar um resistor de 220 Ohms e um capacitor de 1 μF (o modelo do capacitor deve estar adequado ao tipo e valor da tensão da carga).

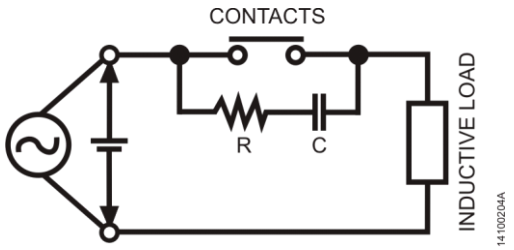


Figura 8-6. CKT RC paralelo com os contatos

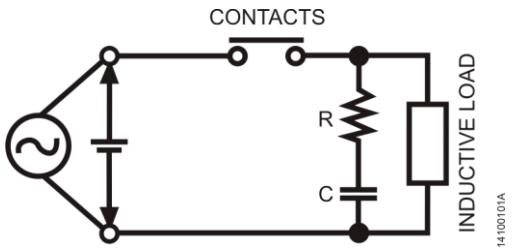


Figura 8-7. CKT RC paralelo com a carga

ATENÇÃO: Para tensões contínuas de 110/125 Vdc, deve ser usado o circuito de proteção RC em paralelo com a carga, sob pena de desgaste acentuado dos contatos. Sugere-se os seguintes valores de R e C: 33 Ohms e 470 μF, respectivamente.

Circuito com Diodo:

Esta é a forma mais eficiente para limitar a tensão do circuito indutivo no momento do desarme. Porém, pode trazer problemas pois aumenta o tempo de desarme caso a carga seja, por exemplo, uma contactora ou solenóide.

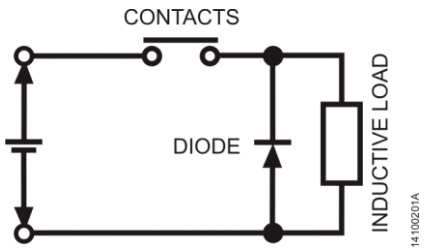


Figura 8-8. CKT com diodo

O circuito pode ser utilizado somente para tensões contínuas, sua tensão reversa recomendada deve ser 10 vezes maior que a da fonte e a corrente no mínimo igual a da carga.

Circuito com Diodo e Zener

O circuito com diodo e zener é adequado quando o tempo de desarme do circuito com diodo é excessivo. Assim como o circuito de proteção com diodo, ele só deve ser utilizado em tensões contínuas. A tensão do zener deve ser um pouco superior da tensão de pico da carga

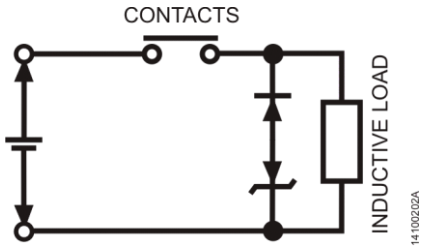


Figura 8-9. CKT diodo zener

Circuito com Varistor

O circuito com varistor limita a tensão do circuito indutivo de forma semelhante a um diodo zener. Sua tensão de condução é em geral maior que um diodo zener e é bidirecional, possibilitando seu uso em circuitos DC ou AC, onde é mais utilizado. Deve ser selecionado conforme tensão máxima da fonte, energia armazenada na carga e vida útil desejada.

ATENÇÃO: É imprescindível a leitura completa das características técnicas dos produtos utilizados antes da instalação ou utilização dos mesmos. Para a escolha do tipo de supressor a ser utilizado, deve-se verificar o tipo de carga (DC ou AC) e os níveis de tensão suportado pelo módulo escolhido para o projeto

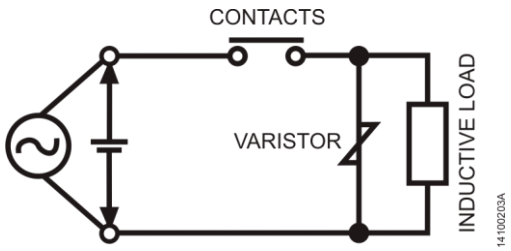


Figura 8-10. CKT Varistor

Circuito com Capacitor

O circuito com capacitor é altamente eficaz para suprimir arcos gerados durante a abertura dos contatos, mas pode causar desgaste do contato em função da corrente para carga e descarga do capacitor. Para a seleção do capacitor usar a mesma regra do circuito RC.

Este circuito é desaconselhado na maioria das aplicações. Somente deve ser escolhido quando os circuitos anteriores se mostrarem inadequados.

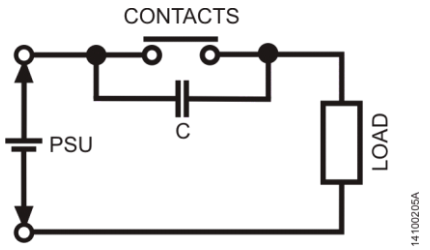


Figura 8-11. CKT com capacitor em paralelo com contatos

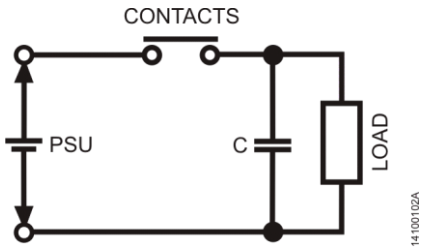


Figura 8-12. CKT com capacitor em paralelo com a carga

Distribuição das Alimentações fora do Armário

Em aplicações onde o armário está distante da máquina ou do sistema a ser controlado, embora esteja no mesmo prédio, recomendam-se os seguintes procedimentos:

1. A condução dos cabos do armário à máquina deve ser feita em condutores metálicos
2. O aterramento destes dutos deve ser feito a cada 20 metros
3. Separar os cabos em dois grupos para distribuição nos dutos:
4. Cabos de sinais digitais até 60 V, cabos blindados conduzindo sinais analógicos e cabos blindados com alimentações até 230 V
5. Cabos com tensão superior a 230 V

Proteção contra Raios

Em aplicações externas, ou seja, em que os cabos ou linhas de comunicação do UCP com os sinais de campo saiam para fora da instalação ou percorram caminhos a céu aberto, deve-se considerar os possíveis danos causados por raios.

Recomenda-se o uso de varistores ou arrestores (com gases inertes) nestes cabos, para proteção do sistema contra sobre tensões decorrentes da queda de raios nestas linhas. Algumas blindagens também são necessárias, conforme mostra a Figura 8-13.

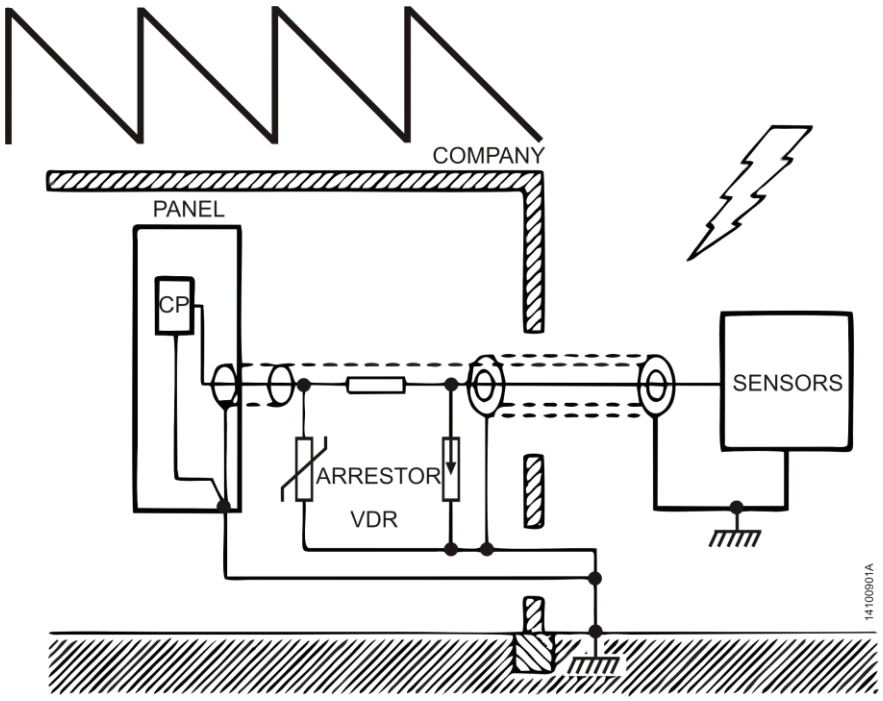


Figura 8-13. Proteção Contra Raios

É recomendável que se instalem estes dispositivos de proteção junto à entrada da indústria ou mesmo junto ao armário.

A Figura 8-13 mostra a forma correta de instalação de proteção contra raios para um sistema genérico. Cada sistema possui detalhes próprios de instalação, portanto recomenda-se que se estudem cada caso individualmente para definição da melhor forma de proteção.

Em casos considerados críticos, consulte diretamente o serviço de suporte da Altus.

9. Glossário

Barramento	Conjunto de sinais elétricos agrupados logicamente com a função de transferir informação e controle entre diferentes elementos de um subsistema.
Bastidor	Plataforma metálica com inteligência lógica onde serão inseridosos módulos para uso no barramento.
Baud rate	Taxa com que os bits de informação são transmitidos através de uma interface serial ou rede de comunicação (medido em bits/segundo).
Bit	Unidade básica de informação, podendo estar no estado 0 ou 1.
Broadcast	Disseminação simultânea de informação a todos os nós interligados a uma rede de comunicação.
Byte	Unidade de informação composta por oito bits.
Canal serial	Interface de um equipamento que transfere dados no modo serial.
Cão-de-guarda de Hardware	Circuito eletrônico destinado a verificar a integridade do funcionamento de um equipamento.
COM	Interface física da UCP de comunicação de protocolos via serial
Controlador programável	Também chamado de UCP. Equipamento que realiza controle sob o comando de um programa aplicativo. É composto de uma UCP, uma fonte de alimentação e uma estrutura de E/S.
Diagnóstico	Procedimento utilizado para detectar e isolar falhas. É também o conjunto de dados usados para tal determinação, que serve para a análise e correção de problemas.
Download	Carga de programa ou configuração na UTR.
E/S	Veja entrada/saída.
EIA RS-485	Padrão industrial (nível físico) para comunicação de dados.
Endereço de módulo	Endereço pelo qual a UTR realiza acessos a um determinado módulo de E/S.
Entrada/saída	Também chamado de E/S. Dispositivos de E/S de dados de um sistema. No caso de CPs, correspondem tipicamente a módulos digitais ou analógicos de entrada ou saída que monitoram ou acionam o dispositivo controlado.
Escravo	Equipamento ligado a uma rede de comunicação que só transmite dados se for solicitado por outro equipamento denominado mestre.
Frame	Uma unidade de informação transmitida na rede.
Full Duplex	Indica que os dispositivos podem realizar comunicações transmitindo/recebendo dados, em ambos os sentidos simultaneamente, ou seja, podem transmitir e receber ao mesmo tempo.
Gateway	Equipamento para a conexão de duas redes de comunicação com diferentes protocolos.
GVL	Global Variable List, objeto onde são declaradas as variáveis globais utilizadas na aplicação.
Half Duplex	Indica que os dispositivos podem realizar comunicações transmitindo/recebendo dados, mas somente em um sentido de cada vez, ou seja, podem transmitir ou receber dados.
Hardware	Equipamentos físicos usados em processamento de dados onde normalmente são executados programas (software).
HSDN	High Speed Deterministic Network. Rede determinística de alto desempenho.
IEC	Sigla para International Electrotechnical Commission, ou Comissão Eletrotécnica Internacional, é um órgão internacional de normalização que prepara e publica normas internacionais no âmbito elétrico, eletrônico e de tecnologias relacionadas.
IEC 61131	Norma genérica para operação e utilização de CPs. Antiga IEC 1131.
IEC-61131-3	Terceira parte da norma genérica para operação e utilização de CPs, IEC61131.
IEC-61850	Norma para subestações elétricas e sua automação,o Protocolo de comunicação refere-se a esta norma, levando o mesmo nome.
Interface	Dispositivo que adapta elétrica e/ou logicamente a transferência de sinais entre dois equipamentos.
Interrupção	Evento com atendimento prioritário que temporariamente suspende a execução de um programa e desvia para uma rotina de atendimento específica
I/O Driver	É um driver de comunicação que possui entradas e saídas entre um bloco/função e outro, por exemplo MODBUS I/O Driver.
IRIG-B	Protocolo de sincronização de tempo utilizando sinal de GPS.
Kbytes	Unidade representativa de quantidade de memória. Representa 1024 bytes.
LED	Sigla para light emitting diode. É um tipo de diodo semicondutor que emite luz quando estimulado por eletricidade. Utilizado como indicador luminoso.
Linguagem de programação	Um conjunto de regras e convenções utilizado para a elaboração de um programa.
Lógica	Matriz gráfica onde são inseridas as instruções de linguagem de um diagrama de relés ou blocos funcionais que compõem uma aplicação. Um conjunto de lógicas ordenadas sequencialmente constitui uma unidade de organização de programa, ou seja, uma subdivisão da aplicação.
Logical Node	É a menor parte de uma função que troca dados e representa uma função num dispositivo físico.
MasterTool Xtorm	Identifica o programa Altus para microcomputador, executável em ambiente Windows®, que permite o

	desenvolvimento de aplicativos para as UTR´s da Série Hadron Xtorm. Ao longo do manual, este programa é referido pela própria sigla ou como programador MasterTool Xtorm.
Mapeamento	Endereçamento de uma variável a uma ação, protocolo, etc.
Menu	Conjunto de opções disponíveis e exibidas por um programa no vídeo/visor e que podem ser selecionadas pelo usuário a fim de ativar ou executar uma determinada tarefa.
Mestre	Equipamento ligado a uma rede de comunicação de onde se originam solicitações de comandos para outros equipamentos da rede.
Módulo (referindo-se a hardware)	Elemento básico de um sistema completo que possui funções bem definidas. Normalmente é ligado ao sistema por conectores, podendo ser facilmente substituído.
Módulo (referindo-se a software)	Parte de um programa aplicativo capaz de realizar uma função específica. Pode ser executado independentemente ou em conjunto com outros módulos, trocando informações através da passagem de parâmetros.
Módulo de E/S	Módulo pertencente ao subsistema de entradas e saídas.
NET	Interface física da UCP de comunicação de protocolos via ethernet
Nó	Qualquer estação de uma rede com capacidade de comunicação utilizando um protocolo estabelecido.
Operandos	Elementos sobre os quais as instruções atuam. Podem representar constantes, variáveis ou um conjunto de variáveis.
Protocolo	Regras de procedimentos e formatos convencionais que, mediante sinais de controle, permitem o estabelecimento de uma transmissão de dados e a recuperação de erros entre equipamentos.
POU	Program Organization Unit, ou Unidade de Organização de Programa, é uma subdivisão do programa aplicativo que pode ser escrito em qualquer uma das linguagens disponíveis.
Programa aplicativo	É o programa carregado em uma UTR, que determina o funcionamento do sistema.
Rede de comunicação	Conjunto de equipamentos (nós) interconectados por canais de comunicação.
Rede de comunicação determinística	Rede de comunicação onde a transmissão e a recepção de informações entre os diversos nós é garantida com um tempo máximo conhecido.
Rede de comunicação mestre-escravo	Rede de comunicação onde as transferências de informações são iniciadas somente a partir de um único nó (mestre da rede) ligado ao barramento de dados. Os demais nós da rede (escravos) apenas respondem quando solicitados.
Rede de comunicação multimestre	Rede de comunicação onde as transferências de informações são iniciadas por qualquer nó ligado ao barramento de dados.
RX	Sigla usada para indicar recepção serial.
Sistema redundante	Sistema que contém elementos de reserva ou duplicados para executar determinada tarefa, que podem tolerar determinados tipos de falha sem que execução da tarefa seja comprometida.
SNTP	Simple Network Time Protocol. Protocolo para sincronização de tempo via rede.
Software	Programas de computador, procedimentos e regras relacionadas à operação de um sistema de processamento de dados.
Subrede	Segmento de uma rede de comunicação que interliga um grupo de equipamentos (nós) com o objetivo de isolar o tráfego local ou utilizar diferentes protocolos ou meio físicos.
Tag	Nome associado a uma variável ou a uma lógica que permite uma identificação resumida de seu conteúdo.
Time-out	Tempo preestabelecido máximo para que uma comunicação seja completada. Se for excedido procedimentos de retentiva ou diagnóstico serão ativados.
Troca a quente	Procedimento de substituição de módulos de um sistema sem a necessidade de desenergização do mesmo. Normalmente utilizado em trocas de módulos de E/S.
TX	Sigla usada para indicar transmissão serial.
UCP	Sigla para unidade central de processamento. Controla o fluxo de informações, interpreta e executa as instruções do programa e monitora os dispositivos do sistema.
UCP ativa	Em um sistema redundante, a UCP ativa realiza o controle do sistema, lendo os valores dos pontos de entrada, executando o programa aplicativo e acionando os valores das saídas.
UCP inoperante	É a UCP que não está no estado ativo (controlando o sistema) nem no estado reserva (supervisionando a UCP ativa). Não pode assumir o controle do sistema.
UCP redundante	Corresponde à outra UCP do sistema, como, por exemplo, a UCPB em relação à UCPA e vice-versa.
UCP reserva	Em um sistema redundante, é a UCP que supervisiona a UCP ativa, não realizando o controle do sistema, mas estando pronta para assumir o controle em caso de falha na UCP ativa.
Unidade Terminal Remota	Também chamado de UTR. Equipamento que realiza controle sob o comando de um programa aplicativo. É composto de uma UCP, uma fonte de alimentação e uma estrutura de E/S.
Upload	Leitura do programa ou configuração da UTR.
Variável de Representação Direta	A variável pode ser acessada diretamente na memória, utilizando o endereço desejado. Por exemplo: %QB0.
Variável Simbólica	Variáveis IEC criadas em POUs e GVLs durante o desenvolvimento do aplicativo, as quais não são endereçadas diretamente na memória.
WD	Sigla para cão-de-guarda em inglês (watchdog).
Word	Unidade de informação composta por 16 bits.
Tempo de execução	Veja Tempo de Ciclo.

Tempo de Ciclo	É o tempo que a UCP leva para executar uma determinada tarefa da aplicação.
Intervalo	Define de quanto em quanto tempo uma determinada tarefa será executada.
LCD	Sigla para Liquid Cristal Display ou Display de Cristal Líquido.
RS-232C	É um padrão para troca serial de dados entre dois pontos (ponto a ponto)
RS-422	É um padrão para troca serial de dados entre dois ou mais pontos (ponto a ponto full duplex)
RS-485	É um padrão para troca serial de dados entre dois ou mais pontos (ponto a ponto half duplex).
10Base-T	Tipo de nível físico para redes Ethernet, definida pela norma IEEE 802.3i em 1990. Suporta baud rates de 10 Mbps sobre dois pares de fios trançados de categoria 3.
Categoria 5	Uma das categorias do cabo UTP: par trançado não blindado com 100 ohms de impedância e características elétricas que suportam frequências de transmissão de até 100 MHz. Definido pela norma TIA/EIA 568-A, pode ser usado em redes 10Base-T e 100Base-TX, entre outras.
ScTP	Sigla para screened twisted pair. Mesmo cabo que o UTP, porém todos os pares de fios estão envoltos por uma folha metálica, ou por uma tela metálica trançada, a fim de minimizar a radiação e a susceptibilidade ao ruído externo. Também é conhecido por sUTP (Screened Unshielded Twisted Pair) ou FTP (Foil Twisted Pair).
UTP	Sigla para unshielded twisted pair. Tipo de cabo formado por pares não blindados de fios trançados. Para aplicações de rede, o termo UTP geralmente se refere ao cabo de 100 ohms, categoria 3, 4 ou 5, especificado pela norma TIA/EIA 568-A. Normalmente o cabo UTP possui 4 pares de fios trançados dentro da mesma bainha (encapsulamento externo).
Wizard	É um template iniciado automaticamente ao criar um projeto no ambiente MasterTool para criação de projetos com definições pré configuradas.